Pekka Ketola

Integrating Usability with Concurrent Engineering in Mobile Phone Development

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF TAMPERE

A-2002-5

TAMPERE 2002

Pekka Ketola

Integrating Usability with Concurrent Engineering in Mobile Phone Development

ACADEMIC DISSERTATION

To be presented, with the permission of the Faculty of Information Sciences
of the University of Tampere, for public discussion in
the Paavo Koli Auditorium on June 10th, 2002, at 13.

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF TAMPERE

A-2002-5

TAMPERE 2002

Supervisor: Professor Pertti Järvinen
Department of Computer and Information Sciences
University of Tampere
Tampere, Finland

Opponent: Professor Timo Jokela
Department of Information Processing Science
University of Oulu
Oulu, Finland

Reviewers: Professor Philip Gray
Department of Computing Science
University of Glasgow
Glasgow, Scotland

Doctor Simo Säde
NETikos Finland Oy

Department of Computer and Information Sciences
FIN-33014 UNIVERSITY OF TAMPERE
Finland

Elecronic dissertation
Acta Electronica Universitatis Tamperensis 185
ISBN 951-44-5382-4
ISSN 1456-954X
http://acta.uta.fi

ISBN 951-44-5359-X
ISSN 1457-2060

Tampereen yliopistopaino Oy
Tampere 2002

Ketola, Pekka (2002)
Integrating Usability with Concurrent Engineering in Mobile Phone Development
Department of Computer and Information Sciences, University of Tampere, Finland

Abstract

The technical complexity of mobile phones is continually increasing due to the introduction of new functions and technologies. The development organisations face an increasing challenge in making the products usable and useful. If for any reason an organisation does not apply user-centred design during the product development phase, the usability engineers may encounter problems in doing their daily work efficiently and effectively. A practical solution is to adapt usability engineering to the specific product development process by applying standard project practices, planning and risk management. The problem studied is how usability engineering can be integrated with the Concurrent Engineering development process. The main research activity was to perform usability engineering in a product development lifecycle during the period 1998-2002. A secondary activity was to observe how successful the same usability engineering approach was in parallel development projects.

The main result of this case study is to amend and supplement Concurrent Engineering with usability engineering activities, reported in the form of Usability Engineering Guidelines. They enable effective and efficient usability engineering in a complex product development setting.

The particular activities and products of this study are Usability Assessment, Usability Plan, and Usability Risk Management. Usability Assessment verifies that the development team has a common understanding in a very early phase about the challenges for product usability. On a practical, level the Usability Plan identifies what factors are important for the success of the developed product and by what coordination and execution activities the success (meeting user requirements) can be verified. Usability Risk Management is a method for managing and communicating the emerging usability problem areas during the development process.

This dissertation presents Usability Engineering Guidelines for avoiding and minimising the basic usability engineering problems: lack of management support and usability activities undertaken too late. From the product development perspective, this dissertation shows that usability engineering has an important role in minimizing uncertainties in product design, building and communicating the overall understanding of the product to be developed, and in connecting development teams of parallel design areas. The long-term planning of usability work is a powerful approach especially in organizing and integrating usability work in a complex organisational setting.

Keywords: user-centred design, usability engineering, Concurrent Engineering, smart product, information appliance, mobile phone, product development, action research, innovation.

CR Categories and Subject Descriptors:

C.3	Computer Systems Organization - Special-purpose and application-based systems: Real-time and embedded systems
D.2.9	Software - Management: Life cycle: Productivity, Software process models
H.5.2	Information Systems – Information Interfaces and Presentation- User interfaces: Ergonomics, Evaluation/methodology, User-centered design

ISBN 951-44-5359-X
ISSN 1457-2060

Contents

List of publications

The thesis is based on the following six articles, which will be referred to in the text by their Roman numerals.

I Ketola P. , Hjelmeroos H. and Räihä K.J. (2000). *Coping with Consistency under Multiple Design Constraints: The Case of the Nokia 9000 WWW Browser*. In Personal Technologies Vol. 4 2000. Pp. 86-95. Springer London.

II Ketola P. (2000). Concurrent Usability Engineering. In McDonald S., Waern Y. and Cockton G. (Eds.): *People and Computers XIV - Usability or Else, Proc. of HCI 2000*. Sunderland, U.K., Springer. Pp. 149-161.

III Ketola P. and Röykkee M. (2001). Ergonomy and Usability Factors In a Mobile Handset. In Nygård C.-H., Luopajärvi T., Lusa S. and Leppänen M. (Eds.): *Proc. NES 2001, Promotion of Health through Ergonomic Working and Living Conditions*, 33rd Annual Congress, 2-5 September 2001, Tampere, Finland. Pp. 240-243.

IV Ketola P. (2001). The Usability Plan as Part of the Concurrent Engineering (CE) Process: An Empirical Study. In Callaos N., Audestad J.A. and Sanchez M. (Eds.): *Proc. World Multiconference on Systemics, Cybernetics and Informatics, SCI 2001, Volume IV Mobile/Wireless Computing*. IIIS. Pp. 225-230.

V Ketola P. (2002). What we learned about voice mailbox in Italy? - Experiences about Usability testing at early design phase with low-fidelity prototypes. Accepted for publication in a book to be published by McGraw-Hill in 2002.

VI Ketola P. (2002). Usability Engineering Milestones In Complex Product Development - Experiences At Nokia Mobile Phones. Accepted for publication in the proceedings of IFIP 2002 World Congress (WCC 2002 Stream 9).

Acknowledgements

I want to thank Nokia Mobile Phones for supporting this research in many practical ways, and for making it possible in the first place. The inspiring atmosphere at different parts of the organisation has made my work challenging and rewarding. I am thankful to all my colleagues who have supported me in my research, and who have trialled the Usability Plan in their projects. I am grateful to the whole Nokia 7650 project team for making this research possible. Special thanks go to *Mikko Moilanen* and *Ari Vättö*.

In my research I have got the best possible support and guidance from my supervisor *Pertti Järvinen*. His constructive and innovative way of showing the way forward, providing fast feedback and pointing to fruitful directions has been the key success factor in my research and reporting, for which I want to thank him respectfully.

I wish to express my appreciation to the pre-examiners of this thesis, *Philip Gray* and *Simo Säde,* for their valuable comments and recommendations, and *Joe Cleetus* for useful email discussions about Concurrent Engineering.

Several colleagues have supported me in my work by providing help, comments, coffee, and good criticism, in the middle of busy working days. I would like to thank the persons who have reviewed articles and the manuscript: *Kaisa Väänänen-Vainio-Mattila, John Rieman, Harri Kiljander* - thank you! I also thank my co-writers in two articles, *Heli Hjelmeroos*, *Kari-Jouko Räihä* and *Mika Röykkee* for their co-operation.

My loving thanks go to my family: my beloved wife *Anni,* sons *Lauri* and *Heikki* (who typed their names here by their own hands) and daughter *Helmi*, for being the motivation to work on, and for keeping my feet on the ground during the challenges in combining work and research. Finally, I'm thankful to my parents, *Kirsti and Jouko Ketola*, for all their love and support.

I respectfully want to thank the University of Tampere, Department of Computer and Information Sciences, for publishing this thesis.

Now it is time to look upward and onward. Ultreuya!

Tampere, May 2002
Pekka Ketola

List of Names and Acronyms

CE	Concurrent Engineering
CEP	Concurrent Engineering Process
CPU	Central Processing Unit
CS	Customer Satisfaction
GPRS	General packet radio service
HCI	Human-Computer Interaction
HFE	Human-factors Engineering
HW	Hardware
I(C)T	Information (and Communication) Technology
ISO	International Standard Organisation
IT	Information Technology
MMS	Multimedia Messaging Service
NMP	Nokia Mobile Phones
OS	Operating system
PDA	Personal Digital Assistant
R&D	Research and Development
SIM	Subscriber identity module
SMS	Short Message Service
SW	Software
UE	Usability Engineering
UI	User Interface
WAP	Wireless Application Protocol
WDA	Wireless Digital Assistant
WID	Wireless Information Device
WML	Wireless Markup Language

Key definitions

2G, 2nd generation	2G is the generation of digital networks, communication standards and technologies that enable voice centric mobile communication with basic messaging, data and fax transfer functions (Steinbock xiii).
3G, 3rd generation	3G is a global development of communication standards and technologies that enable the user to access multimedia services with a mobile terminal. It is characterised by the convergence of mobile phones and the Internet, high-speed wireless data access, intelligent networks, and pervasive computing (Steinbock 2001, xxiv).
Attribute	Perceived property of a product (also used by market researchers)
Bluetooth	Bluetooth wireless technology is a worldwide specification for a small-form factor, low-cost radio solution that provides links between mobile computers, mobile phones, other portable handheld devices, and connectivity to the Internet. The specification is developed, published and promoted by the Bluetooth Special Interest Group (SIG).
Cellular network	Mobile network whose working range is covered evenly with the help of the cells of several base stations.
Characteristic	Property of a product, engineering function or parameter
Communicator	A mobile terminal with PDA and data communication functionality.
Complexity	Complexity is difficult to define explicitly, for example, with mathematical clauses. In this study the term is used to address a system that has large number of dependencies, relationships, unknown factors, changing elements and dynamic components, leading to a system difficult to develop, maintain or change.
Copy-product	A new product design based on an existing product.
Criterion	See *Objective*
Design	A process to create a system or a product with functions that meet some needs, the process of devising and laying down the plans that are needed for the manufacturing of a product (Roozenburg & Eekels 1995); the result of a design process
Ergonomics	The scientific foundation, both in terms of data and methodology, for a human-centred approach to design; knowledge and methods that aim to develop product, equipment, furniture and environment to fit the user's capabilities and to promote user safety and health.
Evaluation	Assessing the effect of something, judging or fixing the value or worth of something
Experiment	In an experiment, the investigator sets the users a series of tasks and quantifies their performance
Expert analysis	A specialist of a specific issue makes observations and conclusions
Human-centred design	Designing for users with users; optimising the interactive system of the user, the product and the task in the context of an organisation and the environment.
Human-computer interaction	Human-computer interaction (HCI) is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (ACM 1992).
Interaction design	The selection of behaviour, function, and information and their presentation to users (Cooper 1999, 22).
Iterate	To repeat until a condition is met (repeat x until y)

Iterative design	The procedure in which the fulfillment of requirements concerning a product solution is evaluated and the design is revised until the solution meets the requirements
Measure	To determine by measurement the values of a variable
MP3	MP3 (Mpeg 3) is a compression format used for sound files.
Objective	Related to the requirements of a product, presented as requirement specifications
Objective assessment	Measurement of the interaction between the user and the product objectively, observation of user-product interaction uninfluenced by personal prejudice
Observation	Technique to gather data. Researchers observe (visually) the phenomena of user-product interaction systematically, directly or based on recordings
Participation	User involvement in the design and evaluation of the product with a possibility to influence the decision
Performance	The measurable degree to which a user, product or system performs.
Preference	Preferring, i.e. choosing as more desirable; something that is superior to another item or items, a value
Product	Something sold by an enterprise to its customer (Ulrich & Eppinger 2000); an artifact used by people because of its properties and functions
Product development	A set of activities beginning with the perception of a market opportunity and ending in the production, sale, and delivery of a product (Ulrich & Eppinger 2000).
Property	Actual analysed quality of a product, described as values of actual variables
Prototype:	A demonstrator to be created to represent the product built for testing and experimentation
Reliability	The degree of giving the same result on successive trials; something that is reliable, i.e., can be trusted because it works well
Requirement	A demand applied to the product by the user or someone else
Segmentation	To divide the population into homogeneous groups according to product, user or purchase situation characteristics
Simulation	Initiating representation of equipment, events and task performance
Specification	Description of the characteristics and functions of a product, description of the product requirements to help the design process
Subject	An individual interacting in an experiment, a user in a user trial, an individual involved in some way in a study
Subjective	Dependent on personal beliefs, norms and values
Subjective assessment	The involved user and his/her perception and/of opinion about the target to be evaluated
System	A combination of multiple elements or components
Task	The activities required (used or believed to be necessary) achieving a defined and desired goal; transform the initial state of an object to the final state
Usability	The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (ISO 9241-11 1998)
Usability engineering	Measurement of the usability of a product to find out the needed characteristics of the developed product, measurement and development of usability characteristics vs. requirements
Usability study	Systematic heuristic or experimental evaluation of the interaction between people and the products, equipment and environments they use; cf. user trial

User, end-user, human, citizen, consumer, individual, customer: individual human being interacting with a product or system

User-centred design Human-centred design

User-interface A combination of input and output parts, ways and procedures (technology) through which a user interacts with a product both cognitively and physically; the assessable aspects of the interface between a product and a human

User interface concept An idea of the logic, look and feel of a user interface.

User trial Experimental investigation in which one or more users test versions of a product under controlled conditions (Pheasant 1996), usability trial within research; usability study

Validity The extent to which different variables actually measure what they were intended to measure (Sanders & McCormick 1993); the degree of verification/confirming

Variable cf. property

1 Introduction

Mobile phone manufacturers have continuous challenge in having sufficient human-centred design (HCD) activities as part of development in order to develop usable products. According to Keinonen et al. (1996), even though usability methods exist, there are problems in introducing those into real development organisation. Those problems are caused by the increased separation of users from developers (Sawyer 2001), for example, due to confidentiality of new innovative products or due to product development process that does not leave much space for users to participate in the development process.

Mobile phones have become personal everyday appliances, such as a TV or a watch (Väänänen-Vainio-Mattila and Ruuska 2000), and part of our identity – full of personal meanings and individual experiences (Sacher and Loudon 2002). They go with us from home to work and to leisure places. They enable instant communication whenever needed and storage for personal data. According to Mäenpää (2000, 15) the main function for a mobile phone has changed from communication to life management. The object of this study, a mobile phone development process, leads the reader to think about these high technology devices provoking many kind of emotions, and about highly competing industry. Those readers who own a mobile phone have an understanding about the daily challenges in performing simple tasks, such as phone calls and text input when sending messages, and also some kind of emotional relationship with the very own Phone. Those readers who do not have a mobile phone will be familiar with the object through numerous advertisements and stock reports in media and in everyday situations.

Consumer products, including mobile phones, can be successfully developed technically and sold without special emphasis on interaction design, usability, user needs or human factors. However, the importance and awareness of mobile phone usability has continuously increased during last few years. Reasons for this are the enormous popularity and penetration of mobile phones especially in Europe and Asia (Dataquest 1999), the success of text messaging (Suomen tekstiviestimarkkinat 1998 - 2002, 2001), and the fact that these devices are technically complex, have a lot of user interface functionality in pocketable size, and they should be useful and usable for very different kinds of users. The markets are huge and mobile phone manufacturers are having an intensive battle for gaining larger market shares. Usability, ease-of-use and usefulness are words that are often mentioned as important factors when a customer makes his or her buying decision (Wilson 2001) and even when the future directions of the industry are predicted (Talouselämä 2001, 63)

Mobile phone manufacturers have a difficult task. Competition drives the products to be smaller, have more functionality, provide latest technological inventions and still they should be more usable for all the time expanding user population, special user groups and users with different cultural backgrounds. In competing product differentiation, variables such as quality, functions, and services are critical. Other important variables are product variety, brand name, packaging, size, warranty, and returns (Häikiö 2001, Steinbock 2001, 270). Very few mobile phone users are able to describe even the basic technological concepts, such as the SIM card or mobile phone operator (Kasesniemi and Rautiainen 2001, 76). Late adopters (Rogers 1995), the majority of users, are not interested about technology as such. They just want to use the product to help themselves in everyday person-to-person communication and achieve their goals, instead (Norman 1998, 31-33, Cooper 1999).

A *mobile phone* is the general title used primarily in Europe and Asia for a wireless telephone. In USA these devices are better known as *cell phones* (by customers) or *mobile handsets* (by manufacturers, dealers and operators). In technical writing, for example in standards, the devices are known as mobile terminals. *Smart phone* has originally been used as a generic title for digital mobile phone that has capability to browse the Internet, receive and send faxes, SMS and email. Symbian[1] uses name WID (Wireless Information Device) for smart phones and communicators (Symbian 2000) and Microsoft uses term WDA (Wireless Digital Assistant) (Microsoft 2002). The different terminal concepts are further described in Section 4.1.

A mobile phone, especially a smart phone (Figure 1.1), is technically a challenging product to design and implement. It has, for example, large amounts of software in many levels (digital signal processing software, operating system, application software), tightly integrated hardware (computer electronics, radio unit, microphone) and mechanical parts (input system, display, covers, even integrated digital camera (Nokia 7650)). The technical *complexity* of products is rapidly increasing while new mobile phone generations are born (Boland and Tenkasi 1995).

Main distinctions to other pocketable appliances, such as digital camera, MP3 player and PDA device, are the wireless voice and data communication capabilities in cellular networks.

Figure 1.1. Two examples of Nokia smart phone concepts, and one smart phone product (Nokia 7650).

Nokia and other mobile phone manufacturers are using some form of parallel or **Concurrent Engineering** (CE) product development. CE provides fast implementation time for a new product and optimises horizontal project coordination enabled by electronic workflow (Fulk and DeSanctis 1995). According to Business week (1990), benefits of CE include 30% to 70% less development time, 65% to 90% fewer engineering changes, 20% to 90% less time to market, 200% to 600% higher quality, and 20% to 110% higher white collar productivity.

In order to coordinate and synchronise the several concurrent design areas, the product development needs sequential phases. Development phases are separated by checkpoints (milestones) to ensure an optimal and synchronised development timetable, to minimise project risks and to verify that product integration succeeds according to plans. Milestones are places where the organisation decides whether it is appropriate to continue to next phase. Sometimes the organisation decides to delay the targeted milestone to meet the milestone criteria better. In practice, design compromises need to be done (acceptable design instead of best design) in order to meet the timetable requirements. In this study the generic product design phases are Specification (M0), Design (M1), Implementation (M2), Testing (M3), Product launch (M4) and

[1] Joint venture of Nokia, Ericsson, Motorola and PsionPLC. The objective of Symbian was to develop an operating system that would stop Microsoft of entering mobile phone industry (Häikiö 2001, 214).

Project end (M5) (Table 1.1). M2 is the milestone where the product *design* ends. After M3 there are activities for correcting and adjusting product implementation until the requirements are met.

M0		M1		M2		M3		M4		M5
	Specify		Design		Implement and integrate		Test		Launch	

Table 1.1. Milestones and development phases.

Product development can be studied from different perspectives. The interest of this work is in achieving usability during the technical system development, especially in mobile phone development. We are interested in the organisation that creates the product, organisational environment in which the product is created, product specific technical factors that form the product usability, and finally the methods how usability of a product can be implemented and verified.

The concept of usability needs clarification. **Usability** can be defined technically as a software quality attribute (ISO 9126). In this definition Usability is seen as a set of software *attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users*. It can also be defined from ergonomic point of view as *an extent to which the product can be used by the user to achieve specified goals in a specified context of use* (ISO 13407).

Literature provides several definitions of usability. Shackel (1991, 25) defines usability in terms of effectiveness, learnability, flexibility and attitude. Nielsen (1994) defines the following attributes for usability: learnability, memorability, error prevention and satisfaction. Jordan (1998) completes the list with guessability, learnability, experience user performance, system potential, and re-usability. Learnability is the common element that follows all different definitions of usability.

Usability problems can be seen as design errors. Design errors can be avoided and errors corrected only if they are found early enough, or not generated at all. Technical product testing tries to find out the implementation errors that the *product or prototype has*. Usability engineering tries to find out design errors already in the early product design phase, prevent and eliminate usability problems *that could emerge in the interaction between the user and product*. Usability engineering has its focus, first of all, on the user. The “goodness” of the resulting design is reflected by the usability of the interface and is measured by usability evaluation and testing techniques (Parush 2001).

The context of use consist of the users, tasks and equipment (hardware, software and materials), and the physical and social environment in which the product is used (ISO 9241). In addition to the ISO 9241 definition there is a need to consider also the information environment, such as the (not) available services in a particular place. Wireless services are perceived as most valuable if they resolve an immediate need in an immediate fashion (Sacher and Loudon 2002). In the area of HCI, it has been recognised for many years that the users and the tasks they carry out are likely to have a strong effect on the results of any system evaluation (Miller 1971). Context of use is an important concept, especially for mobile devices, because they are used in different environments. In the worst case, the product may be usable only in a certain range of contexts (Maguire 2001).

The same mobile phone can be used in wide range of contexts, from office environment to hiking at Himalayas, and from simple family communication to hazardous emergency rescue.

The physical as well as the info-psycho-social contexts have a great deal of variation, even within single user. Activity theory (Leontjev 1981) defines the activity hierarchy: activity – task – operation. The relation between the individual user and the artefact involves the relation between the properties of the individual and the properties of the artefact. A holistic approach to usability acknowledges the different relations and the different levels of activities. Although usability may be assessed in terms of, e.g., efficiency, learnability, and satisfaction, it can only be correctly understood if addressed as *a quality which emerges between a user and artefact in relation to a goal in a use context* within the larger socio-technical context (Karlsson 2001).

Usability engineering can be seen as set of methods for achieving a usability goal: a system that provides a usable user interface. It is *measurement of the usability of a product to find out the needed characteristics of the developed product, measurement and development of usability characteristics vs. requirements* (Kirvesoja 2001). Usability engineering produces information and knowledge about the studied system and users, for example, problems that users have with a specific task. The information and knowledge can be used to build and evaluate a system.

Efficient usability engineering is to work in effective and competent manner in the current development phase with little wasted effort. Efficiency can be measured by comparing realisation (output) against usability engineering goals, plans and amount of work (input).

Effective usability engineering aims to produce an adequate or desired result. Effectiveness can be measured by studying how usable the final product is, or by analysing field feedback. During product development the effectiveness can be estimated, for example, by comparing the performed usability work against the number of usability engineering originated design changes. Planning benefits, in general, are difficult to assess using objective measures. Perceptual measures such as fulfilment of planning objectives as measure of planning effectiveness have better success (Premkumar and King 1994).

Usability engineering is efficient when it has capability to cover the needed product development areas with sufficient actions and resources and effective when it helps improving and verifying the product design. Practitioners know by heart that there are many kind of obstacles preventing efficient and effective usability engineering, and those are also discussed in literature. The main problems preventing effective and efficient usability engineering are:

- *organisational and managerial problems*, such as lack of management support and understanding the profits of usability engineering. These are described by Keinonen et al. (1996), Trenner and Bawa (1998), Mayhew (1999), and Raddle and Young (2001).
- the problem of doing *usability engineering too late* is discussed in Dumash and Redish (1993). They address the need to change the *design process* when usability problems are observed too late thus causing high-cost or unimplementable changes. Nielsen (1996) points out that goals of increased usability and decreased development time are conflicting.
- the *cost of design changes* in the design process increases rapidly when the design moves from the early phases to the later refinement and testing phases (Kiljander 1997).

Usability engineering aims at optimising the product for the intended use with usability activities, such as iterative design, and by involving users in the design process. A major motivation for CE is to develop the product in a limited time. Also, an important factor with leading edge product development is to keep the product highly confidential during the product development time. The objectives of these two engineering areas are hence fundamentally conflicting. As a researcher I see here very interesting and challenging problem domain: what are the possibilities for usability engineering in such organisational environment that by default does not follow or is not driven by the principles of human-centred design? Is the organisational change the only alternative or can human-centred design be successfully adapted to CE without restructuring the organisation?

Factors that enable successful usability engineering are (Mayhew 1999):

- established credibility
- effective communication
- usability buy-in
- engineering approach (instead of artistic approach)
- well-defined work products
- managing of expectations
- clear added-value and
- continuous testing

The objective of this study is to examine on one hand what opportunities CE provides for usability engineering, and on the other hand experiment what is the effect of engineering approach and Usability Plan in executing usability tasks in the development lifecycle.

In three mobile communicator development projects I have observed the following. If usability engineering is not performed during the product development process, it has two kinds of consequences. First, the project development team may have insufficient visibility or objectivity towards the usability of the product in development. Secondly, the user may get a product that does not provide quality of use, efficiency or usefulness. In mobile phones typical usability problems are related to techno-oriented implementation of new technology instead of human-centred design (Norman 1998, Väänänen-Vainio-Mattila and Ruuska 2000), ergonomics (for example in design of a small keypad, Wiklund 1987, Hedge 1998) and user interface consistency (Article I). It is not enough to usability engineer only the software User Interface, but the other design areas and product entity need to be part of usability engineering as well.

Successful usability engineering is based on **human-centred design**[2] (ISO 13407), and successful human-centred design is based on technology, marketing, and user experience (Norman 1998, 40).By definition, Human-centred design is a continuous process that starts by studying the users, creates the design by iterative steps until the design is accepted by the users, and ends by having feedback from users when the system or product is ready. It optimises the interactive system of the user, the product and the task in the context of an organisation and the environment (Kirvesoja 2001).

Human-centred design is a good method for building a usable user interface especially when it can be performed as a continuous and whole process. This makes it possible to take advantage of user participation, iteration and field feedback. However, when Human-centred design is applied with CE some challenges are met. A typical CE development project doesn't provide possibilities for some essential actions of Human-centred design and enough time for iterative design. Design iterations and evaluations need to be performed in a limited timeframe due to overall project timetables and product entity is available for field feedback only after the CE project has finished its task, for example (Article II). In addition to that, the product development confidentiality sets typically considerable restrictions for involving user participation from outside the development organisation.

When a new product is developed it is not enough to verify the usability of it only through the software process but it should be studied, tested and improved in all relevant engineering areas. The forming of usability starts already from the product concept and early interaction design.

[2] Also known as User-Centred Design (UCD).

Practically, the software user interface is chronologically the last element where usability can be effected with correcting actions (Article III). In parallel development usability engineering needs to be performed concurrently in several different design areas, for example user interface, software, hardware, mechanics, and product integration. However, in most cases of the design and development, usability is not dealt with at the same level as other aspects of (software) engineering, for example, usability objectives are not set and resources are not given priority by project management (Karat and Dayton 1995).

The challenges for a usability practitioner are to manage usability issues in parallel design activities and their dependencies in order to understand the real functionality of the product design. In the development of large systems it is often impossible to cover the entire product design with available usability resources. Successful usability engineering in such situation requires the same principles as project management: planning the work, prioritising the tasks, managing risks and communicating effectively.

CE development makes Nokia Mobile Phones a *knowledge intensive company* (Boland and Tenkasi 1995) composed of multiple communities with highly specialised technologies and knowledge domains. In a knowledge intensive company it is important to have the ability to make strong perspectives within a community, as well as the ability to take perspective of another into account. In this kind of organisation there is no single person that knows and understands the entire product development, but the understanding is there in the *community of knowing*. Boland and Tenkasi note that it is easy to see the importance of knowledge work in companies involved with new product development in leading edge technologies.

Usability engineering produces such knowledge about the developed product that can be used for project management purposes as well as for product development in different knowledge domains. It can be in an important role in creating larger perspectives towards the product development.

Cleetus[3] (2001) writes: "CE does not have any limitation on the use of customer focus to get the product "right the first time". What CE does is to emphasise that unless the customer requirements are kept in focus *throughout* the project (and not just at the front end when requirements are usually written down), it is unlikely that the end product will satisfy the customer, let alone make them happy." CE process claims to have focus on the customer (for example Cleetus and Reddy 1992). However, no further guidance is given in CE literature *how* it should be done.

Keinonen et al. (1996) reported that even though usability methods exist, there are problems in introducing those into real development organisation. They ask what kind of steps can and should be taken in different situations to attach usability approach as an integral part of development process? Nieminen (1998) proposes that systematic usability engineering can be introduced to product development with small steps, project by project, in order to avoid too big differences to existing practices. Academic research and industrial experience provide a rich view to usability engineering methods, testing experiences and guidelines. However, the problem of planning and managing usability engineering as a project has not been reported. Further, there is very little defined practice or research how a CE product development project can be usability engineered efficiently and effectively.

---

[3] Cleetus K.J. Personal email correspondency. 10.12.2001

1.1 Research problem

> *Process research is concerned with understanding how things evolve over time and why they evolve in this way, and process data therefore consists largely of stories about what happened and who did what when-that is, events, activities, and choices ordered over time (Langley 1999).*

There are critical shortage of empirical studies on information systems problems and constant calls in the scholarly literature for more in-depth process research that will enable us to understand organisational phenomena at more than a superficial level (Langley 1999, Lyytinen 1987). This work tries to provide a piece of information for understanding the nature of CE and related challenges with usability engineering in the development of mobile devices.

The motivation for this dissertation is the daily challenge to do effective and efficient usability engineering in complex and fast mobile phone development and the fact that very little guidance for product usability engineering is proposed in the applied concurrent development process. CE, which is claimed to have focus on the customer, does not have usability engineering elements or tasks. To verify the usability of a specific product, a usability engineer has to do usability engineering based on 1) general usability methods, 2) earlier experiences and knowledge, for example, in the organisational memory and 3) undefined work practices, often without formal support from the development process.

Without proper planning and right-timed usability engineering there is a high probability that product usability engineering is performed randomly or in a reactive "fire-fighting" mode. Reactive usability engineering is characterised by the fact that problems are understood or noticed very late and the effectiveness and efficiency of usability engineering is low. On the other hand, with intelligent usability engineering the work can be performed in an organised proactive way and the whole development team can have advantage of the gained product knowledge already in the early phase of the development. Unlike in other design areas where the concrete development results are seen during the development process, the main benefits of usability engineering are seen not earlier than when the product is delivered to an end user and he or she starts to use it for daily tasks.

Based on the described motivation I have identified a need to study current state and to improve existing usability engineering practices to meet the real life and especially in the context of CE process in mobile phone development. One way to do this is to organise usability engineering to be an integrated element in the development organisation. On the other hand, I'm also trying to study to what extent it is possible to *strengthen and improve* CE process with the help of usability engineering, without taking the painful step of changing established product development practices to human-centred design. The following research problem can be derived:

How can usability engineering be effectively and efficiently performed in a CE mobile phone development project?

[5] Geos is an operating system developed by Geoworks Corporation. Geos OS was used in Nokia 9110 communicator.

In the following, I will present six research topics and the exact research questions in order to solve the general research problem:

- HCI and usability engineering has its origins in software development for desktop computers and interactive software systems. These systems, that are most often operated via large display and QWERTY keyboard in an office-like context, are very different from mobile phones that are typically operated via small display and keypad in a mobile context. Usability is mostly seen as an attribute of *software* functionality and quality. However, mobile phones being in question it is necessary to see them as carefully integrated product entities where the usability is a result of efforts in several design areas and the integration of the designs. Issues such as human factors, ergonomics and especially context of use can't be omitted. A usability engineer needs to understand who are the users, what are their needs and what are the technical design attributes that have an effect on the product usability.

Question 1: What makes usability in a mobile phone?

- Mobile phones can be developed without usability engineering. In some cases, for example in the development of simpler mainstream or copy-products, it is adequate to rely on design engineer's common sense and intelligence instead of serious usability engineering. However, when products are complex and user interface and mechanical components are made by many separate design teams it is likely that certain predictable type of usability problems will appear on the final product. When the complexity of the product increases, the need for usability engineering increases.

Question 2: What problems arise if usability engineering is not part of the development process?

- A mobile phone development project takes typically 2-4 years, has well-defined development phases and requires efficient information management. Usability engineering in such a project can be done in ad-hoc way or in an organised way. Several development projects have shown that ad-hoc way is inefficient both for results and resource use. Since the projects are fast, even hectic, and resources are always limited there is a need to *plan* usability engineering in order to reach a manageable long-term work practice and good results.

Question 3: How can usability engineering can planned for CE process?

- New product concepts are always challenging. The 15-year history of mobile phones shows that new technologies and functionality concepts are continuously introduced and embedded to products every year. Due to the nature of CE development it is necessary to verify the design already in the early design phase (Väänänen-Vainio-Mattila K. and Ruuska S., 2000) when the technology may not be implemented anywhere yet. The iterative design verification and usability testing must be based on simulations, and low-fidelity or high-fidelity prototypes. Especially low-fidelity prototypes, such as paper prototypes, seem to be a reliable and efficient design tools when new interaction concepts and UI styles are being developed. In addition, paper prototypes seem to provide a cost-efficient solution to the need of early user participance in the design process.

Question 4: Can paper prototypes solve the need of early user testing of new interaction concepts?

- In real life, the pace and nature of concurrent product development guides usability engineering, and not vice versa. The efficiency and effectiveness of usability engineering is thus dependent on how well it can be fitted to that pace. The critical points for successful usability engineering are those where the different phases of the CE start and end. A practitioner needs to understand what are those phases and what effect they have on usability engineering.

Question 5: What are the critical points and what is their effect for usability engineering in CE process?

- In an ideal world we can fundamentally change a product development process to full scale human-centred design. In the real world usability engineering is part of the product development with practical limitations given by the product development process. In my view, usability engineering in a CE project can be seen as a subproject that needs to be managed as a project. Usability engineering is multi-disciplinary activity, covering all product design areas, including project management, and can be an important information and knowledge source to each of them. To achieve the target, a usable mobile phone, right-timed management, planning, implementation and evaluation activities are required.

Question 6: What usability activities need to be in the CE process in order to ensure a usable mobile phone?

1.2 Research strategy

1.2.1 Case study and action research approach

This is a case study with action research approach (Coghlan 2001). It is based on six articles (Publications I - VI) that in this thesis are widely extended with more details, discussion and further references to existing research. Each article discusses usability engineering in the development of smart phones at Nokia Mobile Phones. Articles I - III describe the background information of the challenges in developing usable smart phones, and they provide answers to research questions 1 and 2. Articles IV-VI describe the main findings of the case study, providing answers to research questions 3-6.

The work that is represented in the thesis is based on one case project (a smart phone product development cycle at Nokia Mobile Phones) where CE product development is improved by integrating usability engineering activities with the development process. Although some researchers have advised against using single case studies in process research because of the lack of material for replication and comparison, this strategy provides means of deriving insight from a single rich case (Langley 1999).

Due to the long mobile phone development cycle (typically 2-4 years) it has not been possible to include more case projects in this study. However, observing and participating three earlier development projects and applying (testing) the developed guidelines and methods in three parallel mobile phone development projects assess the results of this study (Figure 1.2).

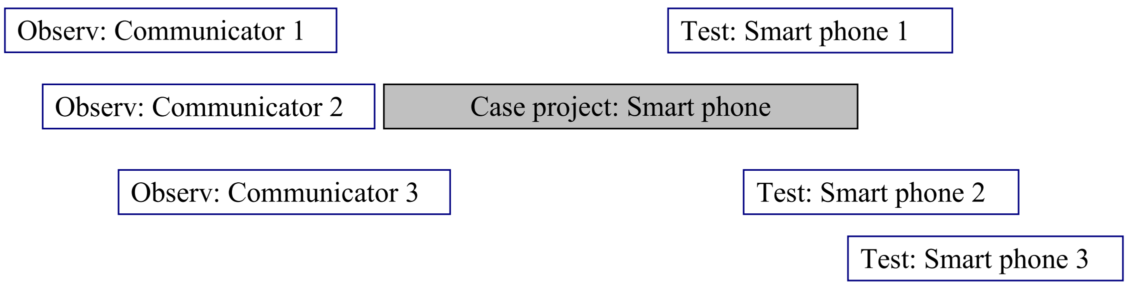

Figure 1.2. Project followed in the study.

During 1995-2001, before this study, I participated in two Nokia communicator development projects as UI designer and usability engineer, and followed the third one from outside during the study. In these projects increasing number of usability activities were made reflecting the increasing organisational interest and capability in usability engineering. My primary focus in these projects was the design of Internet applications, especially WWW browser. These projects form my background information and knowledge about product development at Nokia. During the projects I collected data about the success and problems in product development and in usability engineering, and about the projects as social communities. The knowledge and insights were documented and used later as lessons-learned data in the case project.

Communicator 1 project introduced a new product concept based on an operating system not seen before in mobile devices (Geos[5]) with large number of innovative functions. The new concept integrated a mobile phone and PDA functionality, voice and data communication, and also introduced a new UI style. The main learnings during this project were about:

- the creation of new product concept and category
- working with high confidentiality in product development
- the birth of new large product development organisation
- working with uncertainties in technology development and implementation
- combining features of mobile phones, PDA devices and desktop computers
- dealing with requirements for internal and external consistencies.

Communicator 2 project decreased the size of product, modified most functions of Communicator 1 and introduced several new ones, especially related to data communication. There were several changes also in the physical user interface.

Communicator 3 came out with a new operating system (Symbian) revising practically the whole UI style and all functions of previous communicator products. The three communicator projects and products give us an example of product development that is based on continuous product evolution, discontinuities (Communicator 2 -> Communicator 3), sometimes with big technology leaps, leading to high increase in product variety. The main learnings during these two (Communicator 2 and 3) projects were about:

- using field feedback from the first product on the design of second product
- the evolution of large but young development organisation
- making cost-efficient UI and usability improvements on existing functionality
- embedding new functions into existing ones
- working with the changing and evolving product platforms (user interface, software, hardware, operating system)
- increasing organisational usability capability.

Based on the lessons-learned and observations I formed the ideas describing how usability engineering ought to be integrated in smart phone development. These ideas were formalised in a repeatable form as Usability Plan. Now these ideas are tested with action-research method in the case project.

The case project has several concurrent product development processes with *parallel dependencies* (Järvinen and Kerola 1999) between design teams (for example between user interface, software, hardware and mechanical design) and common checkpoints (milestones) for synchronising and integrating the product development. The product and development organisation complexity for the case project is an ideal framework to perform usability engineering and to experiment with improvement ideas. It provides a full-scale development concurrency and organisational complexity in the project management, design of user interface, industrial design, software, hardware, mechanics and other design areas. The ongoing shift from

2nd to 3rd generation mobile phones and convergence in cellular network systems set also interesting challenges for the usability of the product in the form of new unexplored technology, such as (Väänänen-Vainio-Mattila and Ruuska, 2000):

- the difficulty of testing in all possible usage contexts
- the difficulty of testing human communication practices, especially when it concerns still-unestablished practices
- the difficulty of testing services that cannot all be known beforehand.

Several different stakeholders are directly involved in the study: project management, designers and engineers of different development areas, technology specialists, requirement engineers, and usability engineers in other projects. These persons participated in usability activities as observers, requested user-related information, or needed the information that was produced in usability tests. From project and product development perspective, the key usability activities in the study are:

- organising usability testing and other user-centred design activities according to Usability Plan and emerging needs
- finding and involving participants for usability testing from inside and from outside the company
- identifying and reporting needs for product design improvements
- propagating the identified improvement needs to designers and management
- managing and coordinating outsourced usability activities

These activities together with the guiding Usability Plan produce a large number of meeting minutes, email discussions, design sketches, paper prototypes, usability test plans, data and reports, change request documentation (a database), and specifications. This concrete data produces the main knowledge source in this study[6].

The project environment and atmosphere is ideal. The project management is motivated and committed to enable usability engineering in the case project. The project objectives with quality and timetable are aggressive. The case project is thus characterised by

- development concurrency
- sequential development milestones and phases
- parallel design and implementation dependencies
- technical and organisational complexity

Design teams act in parallel supplying design outputs. The outputs are then integrated to a product. The research focus is in:

- identifying parallel (concurrent) dependencies in the product development organisation.
- identifying horizontal (lifecycle) dependencies in the product development organisation
- studying and identifying those characteristics of a product development that enable effective and efficient usability engineering
- studying and developing the coordination and performance activities that are needed for effective and efficient usability engineering

In order to gain more understanding of the benefits and problems of the proposed Usability Plan, I have closely followed the success of Usability Plan in three parallel projects and interviewed project managers and designers in different design areas. This has provided such insights and

[6] The data cannot be published in this study due to confidentiality limitations.

development topics that have been possible to identify only by being a member of the organisation and development teams.

1.2.2 Project, product and innovation

The aim of the product development project is to create a new “leading edge” smart phone product with several novel technologies, new form factors and new interaction styles. Key functions of the phone are *quality* with excellent messaging and imaging capabilities, *exciting* and easy-to-use interface, *compact* concept, *attractive* design and *entertaining*. The project creates a new product related to several innovation types (Järvinen and Kerola 1978):

- L: Material or technical innovation.
- E: Human or social innovation
- I: Innovation related to data, information, or knowledge
- R: Innovation related to money

This development project deals especially with innovation types L, E and I. The product enables the users to create and manage personal visual content (digital pictures) in a way that enables easy picture messaging from phone to phone and from phone to network service. The whole product concept practically *combines* several innovation types (L + E + I). Innovation type R (money) is not discussed in the case study.

Product developers and product users can create innovations. However, very often introducing an innovation in the product design increases development uncertainty. The case product integrates existing technologies, for example messaging and imaging, in such way that enable new user communication activities and provide several opportunities for new social (user originated) innovations. Several examples of social innovations are described in Kasesniemi and Rautiainen (2001) with the important observation that mobile phone developers are not able to predict or imagine how some technical functions (for example, phone call) will be actually used. Some user groups (the young) invented unexpected ways to use unanswered phone calls as messages and ways to have dialogue. The driver in this invention was to avoid phone call and messaging costs.

1.2.3 Discussion on the research methods and research strategy

The study covers at least three different disciplines in the wide area of information systems research: human-computer interaction (HCI), development processes and organisational studies. For a HCI practitioner this study can be useful in understanding the organisational context of Nokia product development, and for a person looking product development or processes from non-HCI perspective this study may give an insight of the challenges in incorporating usability activities in the development.

This study starts with long-term insider observation of development practices in three product development projects. The researcher, me, is part of the development organisation with the roles of user interface designer and usability engineer. This study has elements from ethnography (insider observation), but having the intention to change the product development process, this is not ethnography by definition. While the objective of ethnography is to provide an understanding of basic human conditions and the change of these (Finken 2000) without imposing new values, my objective is to understand and improve CE development as the environment for doing usability engineering. This setting provides the perspective of being able to identify continual patterns of thoughts and practices and to look into the relationships between them. This position and insider knowledge may blur the openness towards meanings and values that can be both alternative and familiar. Forsythe (1999, 138) discusses the problems that this kind of research setting may have:

> *"…insider ethnography" takes local meanings at face value, overlooking tacit assumptions rather than questioning them. None of these "ethnographic" studies by informaticians raises problems of epistemology or meaning, although anthropologists working in informatics invariably encounter and report such issues, when doing ethnography in similar settings"*

The rest of this study is action research and fundamentally normative, describing how usability engineering ought to be done in Concurrent Engineering. The objective in this research is to turn direct experience from practice, with the ideas familiar from reflection-in-action (Schön 1983), into a form that makes sense to the academic audience as well. Lyytinen (1987) claims that case studies in action research settings seem to be the only means of obtaining sufficiently data for information systems investigations. Being part of the development organisation and testing ideas for better usability engineering frames this work as a reflective conversation where the practitioner functions as agent and experient (Schön 1983, 163).

My organisational position is in the product development team and intentionally *between* management and design teams but not part of either group (Figure 1.3). This position provides the possibility to observe the product development process, conduct the research and to effect change and improvement at any part of the project organisation (Benbasat et al. 1987), i.e. to try to construct an improvement in CE. This is not a typical position of a usability engineer at Nokia. In most cases usability engineers at product development are part of some specific development team, for example software design team (usability engineer), or part of product management team (usability manager). This traditional setting gives often the limitation of not being able to efficiently propagate or sell ideas to other teams while being a member of another team ("not invented here" -attitude).

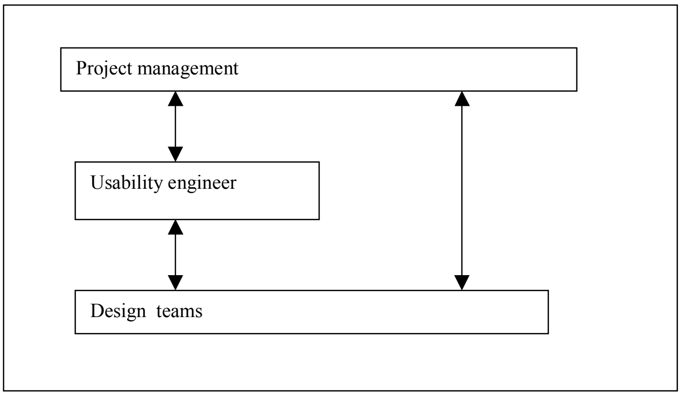

Figure 1.3. Organisational position of the usability engineer in the study.

This study belongs to design science and it is constructive (March and Smith 1995). The objective is to build and evaluate artefacts that help in integrating usability activities into CE process, and evaluate the usefulness of those. The research strategy emphasises the research and learning through intervening and observing the process of change, i.e. building and evaluating sub-processes belong closely to the same process (Järvinen 1999). Action research is a cyclical process and can be described with the following figure (Figure 1.4):

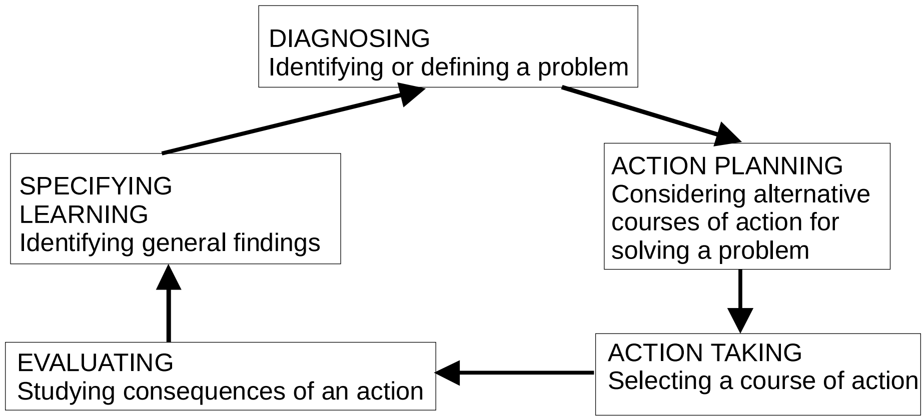

Figure 1.4. The cyclical process of action research (Susman and Evered 1978)

This research is based on the following activities:

1. DIAGNOSING: Observe the existing problems by participating projects where usability engineering is not based on Usability Plan (Communicator 1-3).
2. ACTION PLANNING: Analyse what usability engineering actions would have made those projects more successful.
3. ACTION TAKING: Build a Usability Plan for a mobile phone development project.
4. ACTION TAKING: Execute the plan.
5. EVALUATING: Analyse the success of Usability Plan. This evaluation is needed in each milestone.
6. EVALUATING: Improve and update the Usability Plan based on the findings. This is done with the help of test projects 1-3.
7. SPECIFYING, LEARNING: Analyse the effect of the plan in case project and in other projects.
8. DIAGNOSING: Observe the results of projects that have used a Usability Plan (Future research). New cycle.

These research activities aim to gain a thorough understanding of the application environment, and on the interaction between the technical and non-technical factors of the product development process (Potts 1993). The research activities are described in the following.

Observe the existing problems by participating three mobile phone projects where usability engineering was not systematic. The purpose of observation is to gain insight and understanding of the characteristics and complexity of a mobile phone development project. This part of the study is based on three completed Nokia Communicator development projects. Participation in these successive projects has given hands-on experience on the daily development challenges and information about repeating design and organisational problems. These projects are often characterised with the fact that usability engineering is done mostly in ad-hoc way without an action plan. As a result of the observation, the organisational and design problems that are related to ad-hoc usability engineering are identified. Some of the design problems are reported in article I.

Analyse what usability engineering actions would have made those projects more successful. The goal of this research task is to analyse and define factors that would have made the observed projects more successful in terms of product usability.

Build a Usability Plan for a mobile phone development project. The results of previous task are formalised to a project Usability Plan. The Usability Plan defines coordination tools for usability engineering.

Execute the plan, analyse the success in different development phases. In this research task the effectiveness and efficiency of Usability Plan is analysed through experimentation, observation and evaluation.

Improve and update the plan based on the findings. Based on the observation and evaluation, the Usability Plan is improved and updated for next development phases and for the use of other projects.

Analyse the effect of the plan in other development projects. In this research phase I validate the proposed methodology (Usability Plan based usability engineering) by observing and analysing how it applies to three parallel mobile phone development projects.

1.3 Outline of the Thesis and main results

This thesis studies the challenge of doing successful usability engineering (UE) in a CE project. The approach in this study is to view the product as an integrated entity of hardware and software, and UE as a project that needs to be managed using project management strategies in order to verify the usability of a product. The main results using this approach are:

- Usability Plan is an essential tool for coordinating and executing usability engineering in a CE project.
- product development milestones are important to understand, and they provide predictability for usability engineering tasks.
- human-centred design is possible in a CE project (with certain limitations) even if the product development is not fundamentally based on human-centred approach.
- the main usability engineering problems, lack of management support and usability actions made too late can be minimised with the help of Usability Plan.

Chapter 1 introduces the research object and justification for the research problem. The research strategy is presented and discussed. Chapter 2 describes the related work of this dissertation covering usability engineering, CE and project management strategies. This review shows that there have been attempts to integrate usability engineering with Software Engineering, but no reported efforts about the integration of UE and CE. Chapter 3 describes the organisational environment of Nokia and this study. The concept of CE is studied. Chapter 4 describes a mobile phone development project in order to understand the organisational environment and main dependencies. The main finding is the fact that projects have a lot of common problems, such as product complexity, demanding timetables and resource gaps, and late change proposals in the user interface level are often undoable. This chapter is based on Article II. Chapter 5 defines factors that form the mobile phone usability. The special nature of a mobile phone requires a holistic approach to product usability engineering in order to find the problems in the early design phases. This chapter is based on Article III. Chapter 6 describes usability problems that the final product can have if usability engineering is not effective and efficient. Those problem types are predictable and they are often related to user interface consistency or ergonomics. This chapter is fundamentally based on the findings of Article I. Chapter 7, which is based on Article IV, introduces Usability Plan, a tool for effective and efficient UE. Any systematic engineering is based on a good plan and well-defined work targets. General planning strategies can be successfully applied to UE in wider scale than has been traditionally done. Chapter 8 discusses the product development stages where the usability engineering strategy needs to be changed in order to successfully perform in the changing environment. The changing points follow the development milestones. From UE perspective those points are critical for achieving effectiveness and efficiency. A specific look to the early development and the use of paper prototypes as test tool is studied. This chapter is based on articles V and VI. Chapter 9 proposes guidelines for usability engineering in a CE project with the evaluation of the guidelines. This chapter also provides evidence that the proposed approach is more efficient and effective than competing approaches. Chapter 10 summarises this study, its results, and the limitations of the study, and proposes topics for future research. Chapter 11 provides an introduction to the original articles I-VI, included as appendices.

2 Related work

This chapter provides an overview to the work related to the research on product development, usability engineering lifecycle, usability engineering in product development and usability related organisational issues and standards. In this review I shall show that there have been attempts to integrate usability engineering with software engineering, but integration of UE and CE has not been reported.

2.1 Product development

Steinbock (2001) tells the story of Nokia from 1865 until today (2001). He identifies the success factors and survival strategies of the company that have repeated along Nokia's historical milestones:
- continuous emphasis on innovation, growth and upgrade,
- mastering the full value chain (operations, product development, marketing, sales, service)
- global markets (instead of domestic markets)
- *listening to the customer.*

Customers of Nokia are network operators, service providers, and mobile phone users in low-end and high-end segments. Steinbock (p.304) discusses the problem *which* customers should the company listen to? Steinbock writes "This capability [Listen to the customer] has been perfected into an art through strategy, structure, and resource allocation." Although customer has always been in the product development focus, it is often defined or understood in marketing terms (what functions customers want, who are the target customers), and less often seen as user-centric issue (how the users will use the product, in what context the product is used).

During 1990s the customer-centred thinking has got more emphasis instead of technology-driven product development. Customer fit, customer satisfaction, product segmentation for specific users, user needs and *usability* are continuously repeated by different company managers and leaders. “Focus on customer” is also one of the core ideas of CE (Cleetus and Reddy 1992). Wiklund (1994) gives an overview about organisations succeeded in creating user-friendly products. The successful products are mainly SW systems with an exception from Kodak, the key factor for successful design being design *iteration* with user participation. Gould and Lewis (1991) describe four design principles, normative guides, recommended for system design: early focus on users and tasks, integrated design, early and continual user testing and iterative design. In addition to product development the article also discusses beliefs and organisational issues.

Norman (1998, 216) notes that vertical organisations do not leave place for functions that cut across the structure, such as usability, and they also make the iterative product design difficult. The problem of being able to perform only few design iterations in the short development cycles is also noted by Karat and Dayton (1995). The milestone based product development of NMP does not have process elements for iterative design in the scope of single project, but product development iteration is formed from larger process entities, such as CE, global logistics and customer satisfaction (Valjus 1994). Iteration is not the only solution for efficient product design. Cooper (1999, 242) emphasises the importance of interaction design for shorter development times. With interaction design done in advance, the number of iterations can be reduced significantly.

Concurrent product development is suitable for information appliances due to cost, time and quality efficiency. CE as product development mode was thoroughly studied by independent Concurrent Engineering Research Center (CERC) in West Virginia University starting from August 1988. The objective was to propagate and research CE practices and technology. Research center is still running, but there are not anymore projects focused on CE exclusively. The publications of CERC are numerous and form the core thinking behind CE. The research reports define the concept of CE (Cleetus 1992), principles and metrics for CE (Cleetus and Reddy 1994) and methods to assess the readiness of an organisation to apply CE (Karandikar et al. 1992). Main contemporary CE research is centralized to SCPD (Society of Concurrent Product Development, www.soce.org).

Blackburn et al. (1996) studied how CE principles can be applied in Software Engineering. CE projects often design products where the linkage between software and hardware is tight. The finding in this study is that though concurrency is prevalent *within* product design stages, particularly coding and testing, it is much less prevalent *between* stages. The use of concurrency diminishes as one moves across stages or across the software-hardware interface. However, research on hardware indicates that it is possible to dramatically reduce development time by learning to overlap activities across stages.

Heikkinen (1997) gives a software-focused overview on CE with comparisons to other development modes, such as the Waterfall method. He studied the ways how CE enabling technologies can be used and improved in the development of embedded software systems. The findings show that development process concurrency can be increased by using prototypes and a specific process support environment, leading to better understanding, enhanced communication between development team members and more rigorous project planning and monitoring.

Cleetus (1992) introduces the idea of user (or customer) perspective as one concurrency metric in CE. Nielsen (1996) took this idea forward from usability point of view by studying how system usability can be improved through parallel design. The common goals for both approaches are to reduce the development time and cost and to improve the product quality. Cleetus state that concurrency needs to be applied in all development units, from entities down to specific design tasks: *"A good team leader tries at every stage to partition the work so that many tasks can advance in parallel"*. On the other hand, Nielsen sees parallel design as a method for exploring design alternatives simultaneously. Parallel development has a different meaning for Cleetus and Nielsen.

Tianfield (2001) studies the complexity of product development and proposes a novel advanced life-cycle model for complex product development. Tianfield defines *upstream reach* phase (product development) and *downstream reach* phase (production, use, reuse), and then develops an advanced life-cycle model with the core idea that information-processing dominated processes are first and physical resource-consumption dominated implementations are at the end. The two reasons for this are:

1. Physical resources are limited and, once consumed, irreproducible. However, information is infinitely reproducible.
2. Information-processing dominated processes are less risky in iteration than physical-resource-consumption dominated implementations.

Information-processing dominated processes are cost less in iteration than physical-resource-consumption dominated implementations. Although Tianfield's ideas are not applied or studied in my study, the study provides useful ideas for explaining and understanding *why* product development is complex, what characteristics complex product development has, and what are the development stages in product development. The complexity, he notes, is a result of the product's structure, development organisation and the challenging increasingly critical user requirements. The *usability related* challenges for an organisation are mainly late and difficult-to-identify user requirements. As a result Tianfield states that design and planning of downstream life-cycle stages proceeds and intertwines concurrently with upstream life-cycle stages.

A well-designed user interface is a benchmark for determining the success of a software product. Chao (1993) discusses in a general level the benefits of CE to user interface design. She summarises that CE via teamwork produces consistent interfaces and allows and efficient allocation of techniques, such as rapid prototyping, user involvement and iterative design. Hence, Chao proposes iterative design inside phased CE product development.

Kirvesoja (2001) studies in her dissertation the problem of evaluating usability and ergonomics before the actual product exists, i.e. product concept or prototype. Problems are caused by the many different characteristics and attributes that cannot be measured with the same unit, scale or instrument. As a result she provides procedures that can be easily applied to usability testing in the course of product development.

Work done by Keinonen et al. (1996) is a major inspiration in my work. They studied the problem of designing increasingly complex devices, and discuss the development of smart products presenting ideas for embedding usability in development. The problems for changing existing development practices are, for example, intensity of work, resistance to change and lack of management support. They conclude that usability can be embedded in the product development, but there must be a market-driven or intra-organisational need for change. The need for intra-organisational demand pull is also noted by Kaderbhai (1998). An industry review during 1996 showed that usability is basically a familiar concept but the essential part of usability engineering, user involvement, is still a non-utilized resource (Nieminen and Parkkinen 1998).

Nieminen (Nieminen 1998, Nieminen and Parkkinen 1998) takes a holistic perspective and studies the elements that product development needs to have in order to enable to support the inclusion and management of user related information and activities. The main elements are:

- values and philosophy in development
- generic development stages
- generic development tools and methods
- organisation specific development practices and
- knowledge management.

2.2 Usability engineering lifecycle

The principles of Gould and Lewis (1985) were further developed in Nielsen's *usability engineering* (1994) towards the best-known general formalisation of usability engineering and usability engineering lifecycle. The lifecycle definition provides the following basic phases that should be performed in designing a usable (software) system:

1. Know the user
2. Competitive analysis
3. Setting usability goals
4. Parallel design
5. Participatory design
6. Coordinate design of total interface
7. Apply guidelines and heuristic analysis
8. Prototyping
9. Empirical testing
10. Iterative design
11. Collect feedback from field use

The lifecycle is based on linear steps that are executed in temporal order, i.e. phases, but it has also non-linear iterative activities, such as parallel design, prototyping and iterative design. Järvinen and Järvinen (2000) note that the target of a phased development process is to reach a known target state, while in iterative (evolutionary) process the target state is not fully known but it is defined as part of the design process by iteratively comparing current state to a desired state. Iivari (1987) presents a hierarchical spiral model combining linear development activities with iterative activities. This model is based on baselines and milestones allowing effective status control over the process. Iivari notes that the model helps in directing efforts to the areas in which the uncertainty, and therefore the potential value of the new information, is greatest.

Nielsen's usability engineering lifecycle has been re-formalized in two major publications. ISO 13407 provides a standardized lifecycle containing the same major activities as Nielsen's lifecycle. In addition, it extends field feedback phase in the lifecycle by long-term monitoring. ISO 13407 is supported by other standards related to usability engineering, such as ISO 9241 and ISO 9126. Mayhew (1999) created a widely applied handbook *The Usability Engineering Lifecycle* for usability practitioners describing in a practical way activities how usability engineering can be performed in development. Also her lifecycle definition follows the Nielsen's lifecycle model. The activities are renamed and defined with more details but the fundamental content remains the same. The lifecycle activities by each source are listed in Table 2.1, which shows the correlation between different approaches.

Nielsen	**Mayhew**	**ISO 13407**
Know the user	**Requirement analysis** (User profile, task analysis, platform capabilities /constraints, general design principles, usability goals). Output: Style guide.	Active involvement of users, a clear understanding of user and task requirements
Competitive analysis		Allocate functions between users and technology
Set usability goals		Iterate designs. Multi-disciplinary design
Parallel design		Use existing knowledge for design solutions.
		Understand and specify the context of use. Specify the user and organisational requirements
Participatory design	**Design/Test/Development**: Level 1: Work re-engineering, Conceptual model, CM design, CM mock-ups, iterative CM evaluation. Level 2: Screen design standards, SDS, SDS prototyping, iterative SDS evaluation. Level 3: Detailed UI design, DUID, iterative DUID evaluation.	Use existing knowledge to develop design proposals. Make the design solution concrete using simulations
Coordinate design of total interface		Present the design solution to users and allow them to perform tasks
Apply guidelines & heuristic analysis		Alter the design in response to the user feedback and iterate this process until objectives are met
Prototyping		Manage the iteration of design solutions
Empirical testing		Provide design feedback
Iterative design		Assess whether objectives have been achieved
Feedback from field	Installation (user feedback)	Field validation
		Long-term monitoring

Table 2.1. Usability engineering lifecycles by Nielsen, Mayhew and ISO 13407.

Bevan (1996) provides an overview about the need for applying human-centred design standards in product development. He points out that *usability is the ultimate objective of systems design, which is to achieve quality in use*. Bevan also reports about three ongoing projects (MAPI, INUSE, RESPECT) that aim to incorporate usability and human-centred design into systems development activities in industry.

Several standards and guidelines (for example, ISO 13407, Mayhew 1999, Daly-Jones et al. 1999) define how human-centred design should be performed to provide usable systems. The assumption is that the product development process is based on human-centred design or the desired development mode is human-centred. Hakiel (1997) claims that though we have the knowledge how to do it, the problem is that we do not routinely do what we know.

2.3 Usability engineering in software engineering

> *A design-friendly process is more important than hiring the most talented designer (Cooper 1999, 234).*

The best-known and mostly studied application area for usability engineering is software development, and further, software engineering. Usability engineering is not a process that creates a concrete output, such as software or hardware engineering. Instead it is more like quality engineering having effect on the *quality* of the product. Hakiel (1997) discusses how usability engineering can be integrated with software engineering. He notes that though key software development principles and processes are the same in software and usability engineering, they apply to different domains. In software processes the emphasis is on the quality of code, i.e. defect free code, while the emphasis in usability engineering is in user requirements. Hakiel presents two contrasting approaches to product engineering: (1) usability design deliverables are aligned with software design deliverables(upper part in Figure 2.1) and (2) usability design deliverables are contributing to software requirements (lower part in Figure 2.1). He emphasises the distinction between design for use, which leads to the specification of an information technology artefact, and software development, which leads to the implementation of the artefact in software.

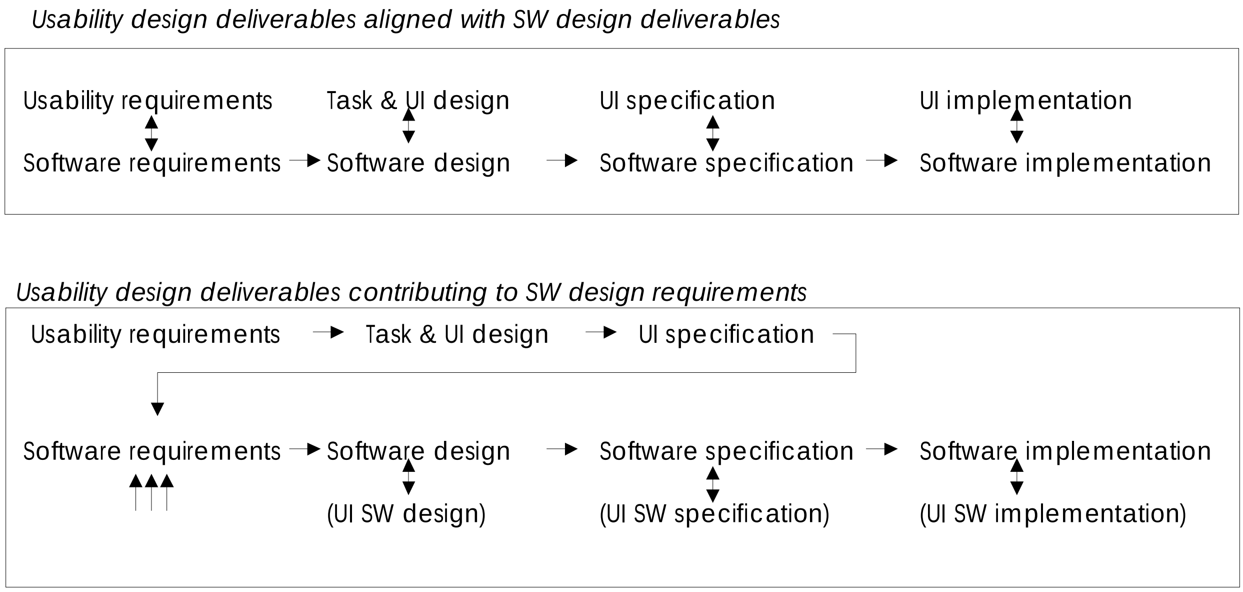

Figure 2.1. Contrasting usability and software engineering approaches.

It would be ideal to perform all design for use before software development. Lyytinen (1987) points out that the idea of a sequential task flow is overly idealistic, and does not correspond to real-life experience.

Usability can be seen as software quality attribute. Karat and Dayton (1995) discuss what kind of education is needed in order to make the organisation understand the ways for achieving quality of use. They describe that companies have difficulties in integrating usability engineering into software engineering for various reasons. Karat and Dayton suggest that in order to enable successful usability engineering (a) formal training and education should consider similarities to other design fields such as architecture and (b) more persons than just usability specialists in the development team needs to be involved in usability activities.

Anderson et al. (2001) describe a case from Shared Medical Systems where human-centred design and usability testing is embedded to software development process instead of handling it as a complementary process in the late phase of the project development. The software development is structured to elaboration, construction, transition and evolution phases. They point out that a high-level agreement is an essential step towards integrating usability into company's process.

Winograd (1995) notes the needed shift from programming environments to design environments (Table 2.2). The interest of software developers is moving away from what the computer does toward the experience of the people who use it. He emphasises the importance of prototypes and understanding the user conceptual model, and proposes several actions for developing better design environments.

Programming Environments	Environments for Software Design
Interactive programming	Responsive prototyping media
Specifications	User conceptual models
Reusable code	Design languages
Interactive debugging	Participatory design

Table 2.2. Suggestive correspondences between current techniques and what is needed for user-oriented software design (Winograd 1995).

Göransson (2001) asked, how to develop usable systems in practice? He, working as a part-time developer / consultant in a company and part-time researcher in an university, applied action research in order to develop Usability design framework for designing usable interactive SW systems. His study is an attempt primarily to "put a face" on human-centred design and try to get organisations and projects to adopt parts of the philosophy behind human-centred design rather than to create new artefacts. Göransson proposes a design process for UCD (Figure 2.2) that follows the ideas of Mayhew (1999) and ISO 13407.

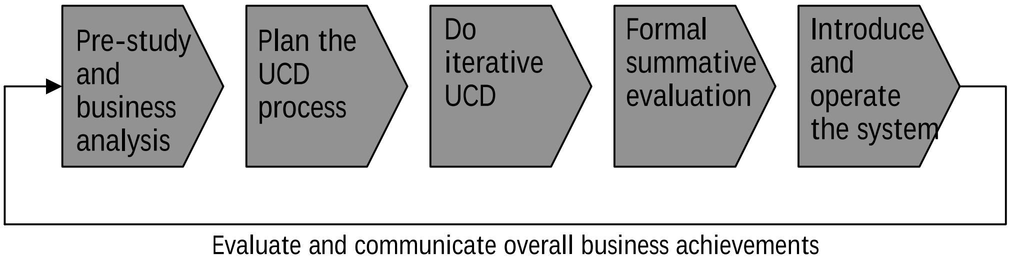

Figure 2.2. Human-centred design lifecycle (Göransson 2001)

Kruchten (2000) discusses why software development projects fail, and lists common symptoms of software development problems. Many of those are directly related to usability engineering, such as:
- inaccurate understanding of end-user needs
- inability to deal with changing requirements
- late discovery of serious project flaws
- unacceptable software performance
- team members in each other's way.

The symptoms are caused by more fundamental problems, the root causes. Kruchten notes that most projects fail because of a combination of the following root causes:
- ad hoc requirements management
- ambiguous and imprecise communication
- overwhelming complexity
- undetected inconsistencies in requirements, designs, and implementations
- insufficient testing
- subjective assessment of project status
- failure to attack risk
- uncontrolled change propagation
- insufficient automation.

2.4 Usability engineering and product development

Usability engineering started as a set of methods for developing usable computer-based software systems, originating from human-factors (ergonomics) research and human-computer interaction research area (Nielsen 1993). Research forums, such as Human Factors and Ergonomics society, originally studied the development of *physically ergonomic* systems. The different approaches are described as follows:

> *Engineering is concerned with improving products from the point of view of mechanical and electrical design, and psychology is concerned with the study of the mind and behaviour. Human factors and ergonomics are concerned with adapting products to people, based upon their physiological and psychological capacities and limitations, the objective being to improve overall system performance (Stanton, 1998).*

Norman (1998, 160) discusses the machine-centred view and human-centred view to product development. He identifies the attributes of humans and machines presented from a human-centred point of view as follows:

People	**Machines**
Creative	Unoriginal
Compliant	Rigid
Attentive to change	Insensitive to change
Resourceful	Unimaginative

In the world around us and appliances in our hands, the software is increasingly embedded to the products and made invisible. The software is an intermediate to operate a device (for example process control of the paper machine) or it is fully invisible (for example TV remote control) to the user. Norman (1998) discusses the problem of intrusive technology in his book *The Invisible Computer*. He claims that many customer products are technology-centred. The problems of technology-centred products are, for example, unusable functions, overwhelming amount of functions, immature technologies and high price. As a promising solution for making simpler products and products that fit to humans Norman proposes human-centred multi-disciplinary product development. A particular method, Contextual design, is proposed.

Most HCI practitioners have to deal with pressured environment of a modern organisation, day-to-day concerns, such as funding, budgets, project and people management, teamwork and communication. Trenner and Bawa (1998) have collected 14 stories from different authors, mostly usability consultants, about the practical organisational problems that usability engineers face in commercial organisations. The main theme in Trenner and Bawa is that in order to perform successful usability engineering you need to have trustable management support. Karat and Dayton (1995) view the same problem from software engineering perspective:

> *In most cases of the design and development of commercial software, usability is not dealt with at the same level as other aspects of software engineering… [Software teams] have had difficulty integrating usability activities into software engineering practice.*

Swanson (1994) notes that the core technology of the business should have certain innovations embedded in order to achieve cost-leadership, differentiation or niche:

- *process* enables better quality and differentiation
- novel *products* and *services* are needed for niche or differentiation

- *integration* enables better coordination and partnership and hence diminishes negative effects of division of labour. Integration is needed also for differentiated products and services.

Often the consumer products that need to be usable are the most personal and most often used products, such as a mobile phone or a watch. A special look at personal information appliances is taken in *Information Appliances and Beyond* (Bergman 2000). This book describes several examples from industry about taking the user-perspective, usability issues and social aspects into account in the product development. A particular value in this book is the formal definition of new concept *Information appliance* and the practical instantiations of the concept:

> *An information appliance is a computer-enhanced consumer device dedicated to a restricted cluster of tasks (p.28)…An information appliance is designed to perform a specific activity, such as music, photography, or writing. A distinguishing feature of information appliance is the ability to share information among themselves (p.3).*

Personal computer is not an information appliance. Some of the key characteristics differentiating information appliances and PCs are (Mohage and Wagner 2000):
- limited purpose and functionality
- not necessarily extensible or upgradeable
- replacement expectation (the user may have to replace the entire device within a few years)
- perceived as less expensive (versus PCs)
- perceived as less complicated to run and maintain (versus PCs)
- very easy to learn and use
- no expectation of "expert users"

Mobile phones have these characteristics. Hence I also consider mobile phones as information appliances. Abowd (2001) notes "*…But a convergent device has to be good at everything it does. It has to be a good phone, a good calendar, a good contact manager, and a good text-messaging/e-mail device. And it has to look nice, feel nice and fit easily in your pocket.*" The focus of usability engineering is inevitably shifting or extending from software development to the development of integrated smart products and information appliances. Product development of information appliances, such as smart phones, is based on fast CE (Valjus 1994). The pace is given by strong competition and business situation in the industry.

Casaday (1996) shares his own ideas and experiences about making usability work in the organisation. He says: "Usability is not a single quality, but it is made up of many goals that can be teased apart and handled somewhat distinctly. It is important to do that so we can focus resources and because *some usability goals are actually in conflict* and have to be traded off." For example, short learning curve tends to be incompatible with very great efficiency (speed), and reliability may conflict with both flexibility and speed. Casaday provides a list of eight usability goals that should be prioritised (1: no importance, 10: essential to product success) in order to guide user interface design. Five of the goals are standard in the usability community, and three are new:

1. Understandable (new)
2. Learnable
3. Memorable
4. Efficient (fast)
5. Reliable (user errors)

6. Flexible (new)
7. Automated (new)
8. Satisfying (subjective)

2.5 Planning and usability engineering

Premkumar and King (1994) studied Information Systems (IS) planning. They provide several findings that can be applied to project planning as well:
- longer planning horizon enables better planning and performance.
- the need for top management and users' continued commitment during implementation are important.
- planning benefits are difficult to assess using objective measures. Perceptual measures such as using the fulfilment of planning objectives as a measure of planning effectiveness have better success.

There are several usability methods covering the planning phase in usability engineering lifecycle, such as competitor analysis, stakeholder meetings, cost-benefit analysis and usability context analysis. They are useful when looking from human-centred design of SW systems perspective but become insufficient when the development domain extends from software development to concurrent development of technical product entities and the perspective is more on the product and project.

> *If the application of human-centred design approach is to be successful, it must be carefully planned and managed throughout all parts of the system development process (Maguire 2001b).*

Mayhew (1999) gives a general-purpose description for usability project planning. According to her, there are several good reasons to have specific Usability Project Plans for product development projects:
- project planning is a standard management technique in most product development projects.
- planning allows to manage work effectively.
- getting the Usability Project Plan included in the overall project plan increases the likelihood that it will be actually executed.
- effective planning and management of usability efforts help in institutionalising usability engineering within the development organisation.

As a conclusion, Mayhew states that it is almost always possible to create a Usability Project Plan that will make a significant contribution to a project.

Daly-Jones et al. (1999) describe Usability planning method in the Handbook of User-Centred Design. The contribution of this method is said to give usability input within a SW engineering programme. Daly-Jones et al. complete the list of planning benefits:
- planning enables priorities to be assessed and facilitates the efficient allocation of resources
- planning ensures that usability work is carried out in a coordinated rather than ad-hoc fashion.
- planning provides clear visibility of what usability work is going on and what the overall aims are.

2.6 Summary

User-centred mobile phone development is full of challenges. We should identify and know our customers, but first we should identify which customer we should listen to (Steinbock 2001, 304). Then we should apply user-centred design (ISO 13407) with usability engineering lifecycle (Nielsen 1993) in technology driven development (Norman 1998, Bergman 2000). User-centred design should be fitted with the CE development environment (Valjus 1994, Heikkinen 1997, Cleetus 1992).

Academic research and industrial experience provides a rich view to product development, human-centred design, usability engineering methods, usability testing experiences and guidelines. However, the challenge of planning and managing usability engineering in concurrent product development has not been well studied nor reported. Further, there is very little defined practice or research for how a complex CE or parallel product development project can be efficiently and effectively usability engineered.

3 Product development and CE at Nokia Mobile Phones

This chapter reviews NMP product development, in particular from design and usability engineering perspective. I shall first describe the main characteristics of Nokia product development (large multi-site organisation, virtual organisation, continuous flow of new products, time-to-market driven product development), and the challenges that are seen in product development. Then I shall study the concept of concurrency and challenge of product development complexity. Concurrency is an essential element of NMP product development and it can be identified in many areas throughout the company.

Mobile phone development is influenced by the special business environment currently dominating mobile communication. This environment can be characterised by (Väänänen-Vainio-Mattila and Ruuska 1999):
- a constant rush and need to bring innovative terminal models to market,
- the speed at which the technology is developing,
- the existence and development of various network standards.

The business environment leads to shortened product development life cycles in order to maintain a competitive advantage.

3.1 Nokia Product Development Organisation

3.1.1 Multi-site product development

Nokia is a global organisation employing 50 000 people world-wide. Products are being developed in 15 countries. NMP, one of Nokia's business units, is focused on developing terminal devices for wireless communication, such as mobile phones, communicators and data cards. The other larger units are Nokia Networks, Nokia Ventures Organisation and Nokia Research Center (Nokia 2000).

Products, sold in 130 countries, are being developed, for example, in Copenhagen, Beijing, Tokyo, Dallas and Helsinki (Helsingin Sanomat 2001). Product development places are called sites. Most development projects are co-efforts between at least two development sites, several projects and numerous subcontractors. Product development at NMP is hence multi-site, multi-project and multi-partner product development. The organisation is a multi-site organisation having global focus towards customers and global mindset in the product development (Steinbock 2001, 136).

Physical distances between development sites vary from few kilometres to several thousands of kilometres. Increased physical distance decreases communication. Hence the challenges of multi-site product development are mostly related to communication:
- different work cultures and ethical values (Steinbock 2001, 210)
- exchange of informal ideas and knowledge, such as coffee room discussions
- co-operation in areas that require face-to-face contacts
- extra work
- exchange of physical objects, such as product prototypes.
- mistrust and tension between people

One of the opportunities for usability engineering provided by multi-site organisation is the instant access to local cultures and first hand knowledge of markets and users near to the development sites. Designers representing different countries and cultures are more able to bring in different points of view and cultural factors than if all the designers were representing only one or limited culture area.

3.1.2 Virtual organization

Virtual organisation can be described with the following dimensions (Nohria and Berkeley 1994, 115):

- electronic files replace material files
- increased computer-mediated communication in primary activities
- increased face-to-face communication in maintaining organisational cohesion
- structure consists of organisation of information and technology rather than persons such that the organisation appears 'structureless'.

Based on the definition above, NMP as a company can be identified as a virtual organisation. Also multi-site product development teams and projects are virtual organisations. The practical challenges of a virtual organisation, very similar to multi-site organisation, are (Fulk and DeSanctis 1995):

- extensive geographic distances
- asynchrony across time zones
- diverse national and regional cultures.

Ideas of Mowshowitz (1997) for virtual organisation reflect the characteristics of product development complexity (Tianfield 2001): "The virtual organisation approach makes explicit the need for dedicated management activities that explore and track the abstract requirements needed to realise some objective while simultaneously, but independently, investigating and specifying the concrete means for satisfying the abstract requirements." In my case project and observed projects, it is possible to identify four basic types of requirements that are related to mobile phone development:

- process requirements. For a virtual organisation it is critical to agree collectively *how* things are done.
- project requirements. For example, when the product must be ready and what resources are given to product development.
- product requirements. Technical attributes for the product.
- user requirements. The tasks the user expects to be able to perform with the product.

Following the definitions of Mowshowitz and Tianfield, a virtual mobile phone development organisation needs dedicated management and coordination activities to handle process, project, product and user requirements.

3.1.3 Continuous and parallel flow of new products

Product development starts from the creation of product concepts. Hundreds of new design concepts are created each year at NMP, but only some of those are developed further to new products (Helsingin Sanomat 2001). With the help of new concepts, NMP is maintaining a continuous flow of new products. Several innovative products and product variants are introduced to customers every year in different market areas.

The challenges of continuous parallel product development and flow are related, for example, to:

- high level management of product portfolio (what products and functions are introduced and when),
- maintaining continuously evolving design heritage,
- introduction of new technologies and functions across products (product consistency),
- maintaining customer satisfaction,
- sufficient support functions for customers, retailers and operators.

A product that reaches the marketplace must fulfill the product and user requirements in order to be successful. When new functions and technologies are introduced with the product it is often difficult to know exactly and in early development phase how users will or want to use it. This was clearly seen in the introduction of mobile phone text messaging. At first it was a little used technical function, but it soon gained huge popularity, especially among certain user groups, and even new communication cultures have born along it (Steinbock 2001, xxvi). Similar challenges and uncertainties are seen in the introduction of embedded entertainment functions (radio, music players) and embedded imaging technology (digital camera).

A special opportunity in continuous product development is the change to iterate design from one product to another and to be able to learn from continuous user feedback. This is important for usability engineering and UI design - areas that should always use the concrete field feedback as design information input. Though the product must be usable in the first product release, “lead product”, there is almost always possibility and most often intra-organisational drive to improve it in the next product if the user requirements are not met.

3.1.4 Time-to-market driven product development

Characteristics for consumer products are decreased product lifespans, the birth of new consumer segments and, hence, increasing number of new products and product variations. A core question for mobile phone development is how to decrease product development time while simultaneously addressing the evolving customer needs (Valjus 1994).

NMP has a special process for defining customer needs and customer satisfaction (CS). In this process the company creates a product *roadmap* for the next 3-5 years. The roadmap defines, among other issues, when specific products are needed and with what functions. Project development projects are started based on the roadmap (Valjus 1994). The roadmap defines the product lifespan (birth and death) for each product. This leads to time-critical product development, time-boxing: a project must deliver the product with required functionality so that it is available when it is needed. Otherwise, there is no business reason to introduce the product. Madachy (2001) discusses the challenges of time-boxing in software development. A general observation is that projects are more frequently becoming constrained by schedule, and delayed because of changing user requirements, unexpected problems and uncertainty in product design decisions. He recommends the use of iterative software development, instead of the Waterfall method, as faster method to develop software systems. However, the schedule is more important for some projects than others.

Several software development methods have been developed to solve the challenge of time-to-market product development and evolving user needs. *Extreme Programming* (XP) (Beck 2001) is a method that combines the ideas of co-operative team-work, implementation of small function sets in time, and function releases. Many of the XP development ideas are also seen in Stepwise Feature Introduction (Back 2002). In this method the software is developed as layers:

> *We consider here an approach to software construction that is based on incrementally extending the system with new features,*

one at a time. We refer to this method as stepwise feature introduction. Introducing a new feature may destroy some already existing features, so the method must allow us to check that old features are preserved when adding a new feature to the system. Also, because the software is being built bottom-up, there is a danger that we get into a blind alley: some new feature cannot be introduced in a proper way because of the way earlier features have been built. Therefore, we need to be able to change the structure of (or refactor) the software when needed (Back 2002).

3.1.5 Concurrent Product Development

An important characteristic and *primum mobile* of NMP product development is the concurrency that can be seen everywhere in the organisation. The core idea of concurrent product development is to propagate the concurrency throughout the organisation and in different levels of product design (Cleetus 1992b). NMP has understood and efficiently implemented this idea. Entire projects work concurrently in order to launch new products in different market areas according to product roadmaps. For each project, the actual product development, sales and production units work concurrently targeting for the same launch day (Lindén 1999). Inside each project, all R&D is done concurrently aiming strictly for the same project milestones. Even inside design teams, sometimes multi-site teams, the product design details are finished concurrently between different team members.

Lindén, research director at Nokia, theorizes (1999) that in a multi-site virtual organisation in principle it is possible to optimize the efficiency of product development concurrency: the work that is started in Japan in the morning, can be continued in Europe or USA when the day goes on. The negative effect of continuous work transfer would be the increased need for job control and coordination while swapping the tasks between teams. Control and job organisation are unproductive tasks that do not contribute to actual product development (Järvinen 1999).

3.2 Concurrent Engineering

In this section I define the concept and implementation of CE. Product development at NMP has been based on CE process since early 1990's (Pulkkinen 1997, Valjus 1994). As a motivation and reason for adopting CE product development Valjus lists the classical product development failures that were observed in linear product development:

- projects were considerably delayed
- the product was not what the markets wanted
- product specification changed too much during project
- manufacturing ramp-up and manufacturing were too difficult
- product quality was not good enough
- company management did not know the real project status.

3.2.1 Definition of Concurrent Engineering

CE was initially developed at 1980s for military purposes at United States (Winner et al. 1988). However, similar approaches have been used for several decades in industry under different names, such as Simultaneous Engineering, Integrated Product Development, Integrated Product and Process Design, etc. (Heikkinen 1997). CE is applied in consumer product development (Valjus 1994) as well as in the development of scientific equipment, such as space vehicles

(Oxnevad 2001). There are two widely referred definitions of CE, both having focus on the customer. The "original" of Winner et al. (1988) and the improved of Cleetus (1992).

> *Concurrent engineering is a systematic approach to integrated development of a product, and its related processes. It emphasises the response to customer expectations and embodies team values of cooperation, trust and sharing. This is done in such a manner that decision making proceeds with large intervals of parallel working by all life-cycle perspectives, synchronised by comparatively brief exchanges to produce consensus (Winner et al. 1988).*

Cleetus (1992) proposes an improved definition for CE by considering what is actually involved in practising it:

> *Concurrent engineering is a* ***systematic approach*** *to integrated development that emphasises* ***response to customer expectation****s and embodies* ***team values*** *of cooperation, trust and sharing in such a manner that* ***decision making*** *proceeds with large intervals of* ***parallel working*** *by all life-cycle perspectives early in the process,* ***synchronized*** *by comparatively brief exchanges to produce* ***consensus****.*

The definitions do not give explicit guidance on applying CE. Thus, CE can be applied in many ways because it is basically a collection of methods, tools and work practices (Heikkinen 1997). Cleetus[7] writes:

> *CE is a way of thinking about any problem, from its definition to is solution, and though all its principles may not have a prominent application in every problem, still it is worth thinking about any problem and keep on asking the questions that a knowledge of CE would prompt you to ask.*

The components of CE can be divided to three categories (Winner et al. 1988, Pawar and Sharifi 1994):

1. Engineering Support Initiatives (Human Factors). These include usage of multi-disciplinary teams, organisational considerations, management support and commitment, availability of skilled people and training, customer focus, and management of the design process.
2. Computer Based Support Initiatives are intended to allow the product development teams to work as effectively as possible.
3. Formal Methods are needed to increase discipline and visibility of the development process.

Heikkinen notes that the key to the CE approach is in the introduction of product teams:

> *Members of the team are drawn from all business functions: marketing, sales, different engineering disciplines, production, after-sales support and quality. Team members meet regularly, and consider the life-cycle of a product at all stages of development. This means that each function can start its work at the earliest possible time, with many activities that were*

[7] Personal correspondence, 10.12. 2001.

traditionally carried out in sequence now being executed concurrently.

CE outcome (product) can be material (such as mobile phones) or immaterial (software). However, CE is highly applicable in such product development where the final product consists of multiple integrated technologies or engineering outcomes, for example integrated software, hardware and mechanics.

By default, CE has focus on the customer. The definition sets the design team's goal in terms of customer satisfaction, rather than in achieving, for example, company proprietary standards. Still, all formal attempts to define CE fail to explain *how* it actually helps in responding to qualitative customers' expectations, such as fulfilling user needs. Current research documentation does not give evidence, with one exception (Chao 1993), that, for example, CE enables better implementation of user requirements or that customers really have participated CE product development. However, there is evidence that customer satisfaction issues are handled *before* and *after* CE process (Valjus 1994), but not during it.

The main objective of CE is to decrease product development time. The secondary objectives are to decrease costs and to provide quality of use (Winner et al. 1988). However, CE can also be applied to such development projects that do no have "users". Decreased product development time is obtained by dividing work to smaller entities that can proceed in parallel instead of advancing sequentially, and by decreasing the amount of research work *in* the project, hence focusing on the actual implementation. Decreased costs are a consequence of development time savings, better teamwork and the capability to avoid costly reworks (Pennell et al. 1989).

CE requires but also stimulates efficient communication between development teams. According to Boland and Tenkasi (1995) innovative product and process creation requires the ability to make strong perspectives within a community, and the ability to take the perspective of another into account.

Though CE improves some aspects of product development, it does not come without problems. The documented problems of CE are related on managing the work:

- difficulties in effective management of teams, team member's limited knowledge, cost of maintaining teams (O'Grady and Young 1991),
- large scale design project is difficult to manage as a whole (Kusiak and Park 1990),
- "missing piece in the puzzle of concurrent engineering is management" (Blackburn et al. 1996).
- there is complicated interaction among concurrent processes and products that may be updated concurrently (Aoyama 1993).
- some activities need to be performed sequentially, i.e. they cannot be split to parallel tasks.
- an activity or a component may have many relations or dependencies to other activities or components. This can be problematic especially in multi-site development.

CE in multi-site organisation is a paradox. On the one hand, CE *enables* multi-site product development by dividing the work on concurrent entities, but on the other hand multi-site development by definition *doesn't encourage* for teamwork:
- CE supports process and job allocation to entities that can be handled independently.
- multi-site product development requires process and job allocation to independent entities.
- CE requires efficient communication between development teams.

- multi-site product development decreases communication between remote teams. Increased physical distance decreases communication.

3.2.2 Concurrent Engineering Product Development Model

Traditional product development models, such as the Waterfall method and spiral models, focus on single sequence of software development process. Concurrent development focuses on the concurrent execution of *multiple* processes (Aoyama 1993).

Figure 3.1 describes a concurrent development process model from Fujitsu. The model is extended with the description of coordination activities to emphasise that entire product development is not possible without project level coordination. The figure demonstrates the existence of multiple parallel development processes. A similar implementation of CE was also realised in this study. This example is a concurrent version of the Waterfall method. According to Ayoama (1993) it is suitable especially to incorporate new product functions and updates into an existing product according to a fixed (cyclic) schedule.

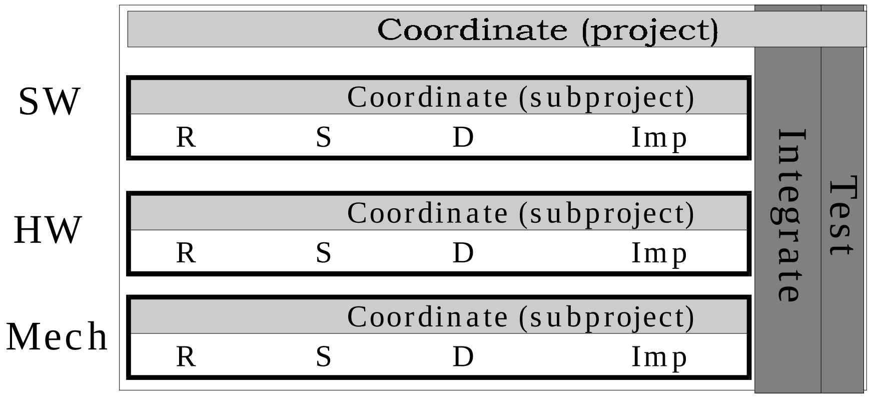

Figure 3.1. Concurrent development process model with coordination activities. Requirement analysis (R), Specification (S), Design (D), Implementation (Imp).

The basic development activities are coordination and execution. Coordination and execution activities are performed in the main process (project level) and in subprocesses (team/subproject level). The subprocess execution process, for example SW engineering, can be independent, i.e. it is not directly dependent on other processes, while the coordination processes are often tightly interconnected, especially in late design and integration phases.

In product development, it is important to identify the difference of coordination and execution activities. According to Järvinen (1980) coordination (control) activities are planning and follow-up. Coordination activities are needed to verify the optimal use of *processes*, navigate the *project* from start to end, for controlling the entity of the developed project and to manage the handling of user requirements. Planning, product design and product testing can be seen as coordination activities for the product implementation.

Execution (implementation) activities are activities that are directed by the coordination activities. For example, SW implementation based on UI specification and appropriate plans is execution activity. Execution activity in the main process (project level) is management. Ayoama (1993) separates *process* management, *project* management and *product* management to separate activities. Process management defines and instantiates development processes, assigns development teams and controls the execution. Project management assigns and controls development resources to ensure productivity and quality of the project. Product management manages the developed product along concurrent processes, starting from requirements specification and design.

The job division, for example to concurrent activities and phases, produces non-productive extra tasks, such as transfer of objects, inspections, checking and communication, coordination, coding, decoding and conversions. Adding concurrency and phases increases the number of non-productive tasks.

From information point of view, the fundamental difference between the traditional sequential development process and the CE approach is in the amount of required co-operation and communication between teams, and inside R&D, design, manufacturing and support activities throughout the development process. In the sequential process the required co-operation is minimal whereas in CE co-operation it is maximal and an explicit action goal.

The engineering practice inside a development team need not be concurrent, but it can follow other more efficient processes as long as common development milestones can be followed. For example, incremental development may be appropriate for software engineering, and simultaneously Waterfall method works better with hardware engineering. Thus, CE is more product development coordination method than product implementation method:

CE is a management technique to reduce the time-to-market in product development through simultaneous performance of activities and processing of information.

With Concurrent product development the organisation is able to run and synchronise multiple parallel design processes in the development of one product. This leads to time savings and potentially cost savings. However, the price for applying CE is in the considerable amount of unproductive coordination and controlling work. The key to successful CE is the emphasis on coordination and project management that is achieved with good process models (Heikkinen 1997, 83) and strict review procedures (Ayoama 1993). The weakness and main risk of CE is in the late integration of product components. By delaying the integration of components to last phases of the development, it is probable that problems appear and large number of software and hardware adjustments needs to be done before the system starts to work properly.

3.2.3 Activity concurrency and information concurrency

Blackburn et al. (1996) separate the two forms of concurrency: concurrency in activities and concurrency in information. *Activity concurrency* refers to the task and design activities that are performed simultaneously by different people or groups. *Information concurrency* refers to the integrated, or team, approach to development in which all the concerns of the different functions - the customer relations, R&D, design, engineering, manufacturing, sales, and service - are addressed through a flow of shared information. Activity concurrency can be seen as a hierarchy in development. The hierarchy ranges from within phase to across projects and platforms.

Within-phase overlap of tasks is the simplest form of activity concurrency. This is commonly practised in detailed design, where the problem of designing is subdivided into modules that are designed in parallel (Figure 3.2). Parallel activity can also be practised within phases, such as the requirements, where different functions work in parallel to specify customer requirements.

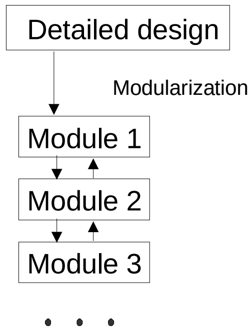

Figure 3.2. Within-phase overlap

Across phase overlap involves concurrent activities across development process phases, such as high-level design, detailed design and testing (Figure 3.3). Instead of waiting until all the design modules are completed before testing, they overlap and time-compress this activity by simulating the presence of other component and incorporating the testing into the detailed design phase. However, the risk and cost of failure are high.

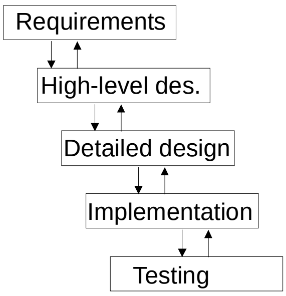

Figure 3.3. Across phase overlap.

Hardware/software overlap occurs when software must be embedded into larger system which integrates hardware and software. Once the high-level design and product specifications have been determined, design of hardware components can be performed in parallel with the software (Figure 3.4).

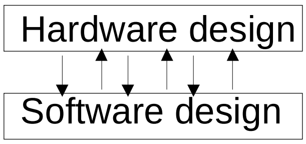

Figure 3.4. Hardware/software overlap

Across project overlap presents a different type of concurrency challenge for management: components designed for one product can be reused in current and future product releases (Figure 3.5). That is, the design activity for two or more products takes place simultaneously through the design of components.

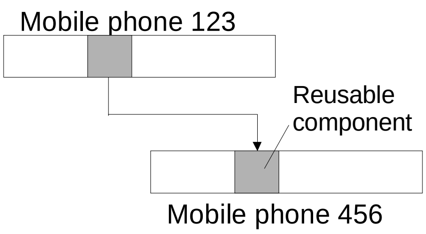

Figure 3.5. Overlap across projects.

Blackburn et al. describe three forms of information concurrency with practical examples: front loading, flying start and two-way high bandwidth flow.

Front-loading is the early implementation of upstream design activities of downstream functions. Front loading provides an early warning about issues that could lead to costly redesign and rework later. In hardware, for example, front loading of manufacturability problems can keep them from becoming show stoppers when the design is transferred to production.

Flying start is preliminary information transfer flowing from upstream design activities to team members primarily concerned with downstream activities. For example, partial design information about components to be used in a product can provide a jump start for process engineering and help compress the time in process design activities.

Two-way high bandwidth information exchange is intensive and rich communication among teams while performing concurrent activities. The information flow includes communication about potential design solutions and design changes to avoid infeasibilities and interface problems. For example, the hardware/software overlap is a situation in which two-way high bandwidth flows are critical: teams involved in concurrent hardware and software design need to have a steady flow of information among the groups to prevent potential integration problems.

Blackburn et al. succeed in capturing the essential elements of CE, especially by separating activity and information concurrencies. However, this is not enough in order to model CE in a multi-site virtual organisation. The missing elements are the physical and temporal distances of people that perform CE.

3.2.4 Metrics for Concurrent Engineering

Metrics of CE should give answer to the question, how good is a particular CE environment. Question "How good?" can be further defined to more detailed questions, such as:

- how well does the CE environment integrate different development activities?
- how effective is the CE environment in terms of cost, time and quality?
- does the CE environment produce good quality products?
- how predictable is the CE environment (cost and time)?

Cleetus and Reddy (1992) propose integration, effectiveness, quality and concurrency measures that can be used to evaluate the CE environment:

- *integration* measures how easily an organisation can develop the inter-operation between engineering tools and CE services, and the simplicity with which engineering user can employ engineering tools and CE services.
- *effectiveness* measures cost and time reduction and quality improvements of products created.
    - *quality improvement* measures the quality improvement *in a given time*. It is perhaps possible to achieve the same quality with sequential approach as that achieved with CE, given a longer time.
- *concurrency* can be measured with four different measures:
    - time-averaged degree of parallel working in different perspectives
    - contribution of a perspective in terms of decisions arrived at or proposed
    - CE process from customer focus perspective
    - current state of decision making in terms of the degree of conflict that persists.

Cleetus and Reddy provide general level ideas of measures. They do not further explain how the proposed measures should be implemented. Even if the measures were well defined, the complexity of CE makes accurate measures difficult.

3.3 Complex product development

Information systems that are developed in time, within a predefined budget, and that meet user's expectations are rare. This is a result of increasing inherent complexity of applications and novel application areas (Hofstede and Verhoef 1997). As we have seen, concurrent product development at NMP is complex. The complexity results from (Tianfield 2001):

- complexity of the product's structure,
- complexity of development organisation and
- complexity of user requirements.

Hofmeister et al. (2001, 27) discusses these two domains, product's structure and organisational factors. The product structure defines the mechanical, logical, and software architecture of the product. It covers issues such as functions, performance, dependability and cost. These factors differ between products of the same company. Organisational factors constrain the design choices, such as schedule and budget. The factors have an effect to every product developed by an organisation. In addition to Tianfield, Hofmeister et al. also discuss technological factors, which define what kind of hardware and software architecture, technology and standards are *currently* available. Hence, technological factors as well as user requirements change over time and products must adapt.

In consumer products there is a strong drive to increase the number of functionality, functions or embedded technologies in a new product. According to Roos and Hoikkala (2000, 13) a mobile phone is like a wheel, which has so much potential for different uses that we are probably not even aware of those yet. In smart products, it is not enough to just add a new function, like a Lego brick, but it often needs dynamic integration to the existing system (Figure 3.6).

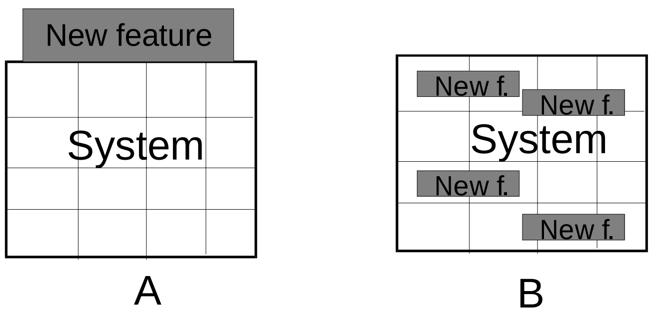

Figure 3.6. Add new function on existing system (A). Integrate new function into existing system (B).

In incremental development (A) new functions do not add the development complexity in the same way as in iterative development (B) because the existing (old) functions may not be changed and new functions do not necessarily change user interaction at all, or the old functionality may be changed only minimally. The introduction of new functions based on the incremental development can be demonstrated with the product definition of Nokia 3xxx phones (www.nokia.com):

Key functions	Nokia 3210	Nokia 3310	Nokia 3330
WAP			X
Chat messaging			X
Animated screen savers			X
Picture viewer			X
Downloadable game packs			X
Vibra function		X	X
Voice dialing		X	X
New games		X	X
Time management		X	X
Picture messaging	x	X	X
Xpress-on™ covers	x	X	X
Predictive text input	x	X	X

Table 3.1. Function introduction in Nokia 3xxx phones.

Introduction of new functions in the incremental way (Figure 3.6, B) demonstrates the more complex way. New technologies are embedded to existing ones, thus requiring new design and implementation in several software components. For example, embedding a digital camera in the product requires enhancements in all the applications that use photos anyhow, such as messaging applications and Phonebook.

The addition of new functions often leads to changes in the user interface and most probably increases the complexity of product from implementation and also from user points of view, i.e. increases the *product variety*, makes the implementation more difficult and may decrease the usability. When new technologies are adopted in the product, the organisation tends to get more complex because implementation of a new technology requires technology specific expertise that may not have been earlier in the organisation. Very seldom, increasing new functions makes the user interface easier to use or the organisation simpler. This can be expressed with the following assumptions:

- development of Product 2 follows (temporally) development of Product 1
- development of Product 2 has more variety than development of Product 1.
- development of Product 2 has more uncertainties than development of Product 1.
    - for example, changed product structure, evolved development organisation, new user requirements, shorter timetable,
    - does not necessarily follow if Product 2 re-uses large amount of Product 1 design and implementation.
- development of Product 2 requires more regulation than development of Product 1 in order to reduce uncertainties.

These assumptions can be formalised as The Law of Requisite Hierarchy:

> *The weaker in average are the regulatory abilities and the larger the uncertainties of available regulators, the more hierarchy is needed in the organisation of regulation and control to attain the same result of regulation, if possible at all (Järvinen 1999, 84).*

The Law of Requisite Hierarchy proposes that while products get more complex (and design uncertainty increases), the capabilities of an organisation needs to be improved. On the other hand, while organisation learns (design uncertainty decreases), the need for organisational change also decreases.

3.4 Design uncertainty

According to Tuikka (1997, 395; 399-401), the design of new concept is characterised by uncertainty about the concept itself, its feasibility, and its relevance as well as by many views of the future product. A new design attribute, *design uncertainty,* can be defined using negation. Design *certainty* describes the extent to which a designer is confident about how a specific artefact should be designed. Design certainty is achieved through training, professional experience, user trials and by learning from earlier designs.

Design uncertainty is the extent to which a designer does not confidently know user requirements or how to design a specific artefact.

If the artefact has a service interface or is in some other way dependent on issues that are not controlled by the development team it is useful to separate

- internal design uncertainty, that deals with practical product design and implementation issues, such as technical limitations, and
- external design uncertainty, that addresses the issues that are dependent on the world outside development organisation. External design uncertainty is caused, for example, by an emerging network technology, or rapidly changing user requirements.

Design uncertainty for a specific artefact leads to questions covering upstream and downstream processes in the product development. Upstream process (product development) questions deal with issues, such as: why this artefact should be designed, how it should be designed, and are we designing a useful or needed artefact or function? Downstream process deals with questions, such as: is this function adding value (to the company or product), who will buy a product with this function, and are we capable to provide support for the users of this function?

Based on the degree of product variety and degree of innovation we can formulate the following assumptions:

1. When product variety increases, the technical product complexity increases (Ashby 1956).
2. When product innovativeness increases, the product design uncertainty increases.
3. When design uncertainty increases, the need for human-centred design increases.

There is often more design uncertainty in innovative products than in products that improve earlier products. The practical consequence is that innovative products are more difficult to design and implement than improving products due to more challenging design uncertainties and technical complexities. The designers of innovative functions are *living in* the world of design uncertainty and they are constantly solving related design challenges.

In the development of innovative products, higher design uncertainties and technical complexities are often seen throughout both the upstream and downstream processes. In products that increase functions to existing products these are seen mostly during early phases of the upstream process in design areas that differ from the earlier products.

3.5 Summary

For a usability engineer, NMP's concurrent and global product development is certainly a challenge. It is not possible to use simple development team-focused engineering strategies, but the usability work needs to take advantage of the richness of the organisational characteristics, complexities and uncertainties. In practice this leads to active networking between usability engineers and designers in other development sites in order to be able to efficiently build and evaluate the developed product.

The timing of the evolution of CE in the United States during the second half of the 1980s was significant for Nokia (Steinbock 2001, 209). CE emerged as a new method of product development capable of being continuously upgraded and modified and was expected to significantly reduce cost and development time without sacrificing quality. Also, in the development of complex consumer products and when the product lifespan is strictly defined, the predictability of product development is a critical issue for the implementation of development roadmaps and timeboxing. The company needed an easy-to implement and easy-to-adapt method with which team members in different geographical locations could work with a unified product concept.

NMP is a complex organisation because of complex product development (The Law of Requisite Hierarchy). Product development is driven by intelligent management and sophisticated business processes. The global business competition will be won by the players that reach the market fastest and more efficiently with products that best fit the customer (Steinbock 2001).

A full implementation of human-centred design (HCD) in CE would provide the ultimate user-satisfaction, but it would require large changes in many parts of the company. The hypothesis and starting points for this study are:

- the product complexity increases continuously,
- the product development (organisation, processes, practices) complexity increases continuously and
- the organisation won't ever fully change to HCD, but it is willing to adopt as much of it as is reasonable due to business reasons.

On the other hand, history of Nokia has shown that the company is capable of implementing fast and large-scale organisational and operational transformations when needed.

A major motivation in using CE is to minimise the product development time and costs. Even though the CE does all this, if well implemented, there are still more efficient ways proposed for product development. Oxnevad (2001) reports that using special Next Generation Project Development Teams (NPDT) it has been possible to shrink development time in the early design phases by factors between four and ten. The key to improved development time is in performing real design analysis and design work during concurrent design sessions and in using high-end analysis tools (prototypes, simulations) in these sessions. The traditional usability engineering is based on the idea where early design is made with low-fidelity prototypes, whereas NPDT prompts for using high-fidelity prototypes in the early design phase.

4 Mobile phone product development

In this chapter I first study the technical concept of a mobile phone and then development project in order to understand the organisational environment and main dependencies *inside* the project. The information in this chapter is based on literature review and learnings from the observed and participated projects. The chapter starts by describing a smart phone. Further on, I study development core processes: sequential development process and concurrent development process. The objective of this chapter is to build a general understanding about the processes that are needed to develop a mobile phone.

NMP's product development can be divided into upstream and downstream processes (Table 4.1), also called Product Development Process and Customer Commitment Process. The core activities in upstream processes are associated with the physical or virtual creation of the product: standards development, platform development, new product development and logistics. Downstream processes deal with branding, product segmentation and design (Steinbock 2001, 130). This chapter focuses on the new product development in the upstream process.

Upstream processes	Downstream processes
- Standards - Platforms - New-product development - Logistics	- Branding - Segmentation - Design
--- Product development process ---	--- Customer commitment process ---

Table 4.1. Upstream and Downstream processes.

From product development point of view, new consumer products can be categorized to two groups:
- products that improve earlier products by introducing predictable functions that exist already in other products, and
- innovative products that introduce functions belonging to one or more of L-, E-, R-, or I-innovation types.

In both groups, the new product changes something that already exists, i.e. the variety in a product increases from earlier products, and the technical complexity of the product increases (the Law of Requisite Variety, Ashby 1956). Innovativeness together with high technical complexity leads to design uncertainty.

4.1 Mobile phone and Smart phone

Gartner (2001) gives a market-based classification of mobile terminals:
- basic phone: A voice-centric device designed to provide only voice functions and limited contact management, as distinguished from an enhanced phone or smart phone.
- enhanced phone: A voice-centric device designed to deliver data content via network-based - e.g., Wireless Application Protocol (WAP) - Web delivery mechanisms that offers only minimal offline capability, such as contact management.
- smart phone: A large-screen, voice-centric handheld device designed to host offline applications, such as a personal information manager.
- wireless information device (WID): An emerging form of mobile computing and communications device - interactive and "always on" in nature. Driven by the coalescence of

a number of technologies, WIDs will evolve from personal digital assistants, Wireless Application Protocol (WAP) telephones and other form factors. Gartner expects that increasing price/performance capabilities of WIDs will combine with improvements in wireless access and user interface technologies to provide inexpensive, lightweight devices that will be broadly adopted and frequently used. Functions such as location sensing and tiny, low-cost cameras will further enhance the range of capabilities and services that users will experience with WIDs.

Smart phone has originally been used as a generic title for digital mobile phone that has capability to browse the Internet, receive and send faxes, SMS and email. Smart phones are voice centric phones offering significant capabilities utilising data management and transfer capabilities. Nokia 9110 and Ericsson 380 are typical examples of smart phones (House 2000). Smart phones can be divided to three levels based on their main characteristics. The levels are presented in Table 4.2.

Level	Description
1	Terminals, which enable Internet browsing and advanced PDA capabilities. Target users: business users with advanced data needs. These terminals are also called *communicators*.
2	Phones with selective Internet access and limited personal functions. This category includes the majority of WAP phones. Target users: larger audience of personal and business mobile phone users who prefer a limited amount of Internet access and PDA functions at an affordable price.
3	Handsets, which have limited data capabilities and limited PDA functions with advanced text and short messaging capabilities.

Table 4.2. Classification of smart phones (The Strategis Group 2000)

The above categorisation of smart phones is communication centric. Browser capability is already becoming a standard function in mobile phones, also in the lower product segments[8]. Nokia Insight (Nokia 2000) provides a similar message - the vision is in "putting the Internet into everybody's pocket."

A finer categorisation of smart phones is emerging. Several manufacturers are providing special phones for imaging, messaging, entertainment or browsing. The smartness is not in communication, but in the way communication capabilities are employed in other functions. Hence, it is not enough to define smart phones based on the browsing or communication capabilities only.

The development of mobile phones and network services creates a possibility to define larger communities based on the idea of mobile communication. The Wireless Village initiative (Nokia 2001) aims to build a community of end users and global business partners where Internet and wireless domains converge. Mobile Information Society is a concept of expected state of communication between people in near future where information flow is based more on mobile terminals rather than wired personal computers. "In information society … citizens are expected to adopt and master new technical skills, the aim being that all or almost all of them should acquire these skills (Statistics Finland 2001)." The mobile information society relies on networks and mobile equipment as a source of services and knowledge to those who are able to benefit them.

[8] For example, Nokia 3330, “a basic phone”, has WAP browser.

The classifications of terminal categories by different parties are slightly in contradiction. This is probably consequence of fast development and introduction of new products and product concepts. Due to the evolution in mobile phone industry we need a better and more customer oriented definition for smart phones:

> *Smart phone* is a digital mobile phone that enables the user to perform daily personal information management tasks, fulfilling the basic human communication needs of a wireless village citizen in the mobile information society.

4.2 Components for mobile phone

In order to understand the challenges that are related to the usability of a mobile phone, we need to understand the main mobile phone design areas that have impact on the user experience. The main areas are industrial design, hardware platform, mechanical concept, software platform, and user interface style. There are also many other design areas, such as localisation, that are left out of this review.

4.2.1 Industrial design

> *"The customer grabs the most attractive product, whatever that means to him or her… Products are no longer bought on the basis of functional values (Pulkkinen 1997, 146)."*

Industrial design has tremendous impact on the product success. For example, watches or scissors are often sold based on the design and not on the functions. A mobile phone is identified and differentiated by its industrial design. Each new mobile phone design is somehow different from earlier designs.

> *"The sizes of mobile handsets are decreasing continuously. New models are lighter, more flat and shorter. The small size and simple-looking face of a mobile phone have been for a long time the main challenges for industrial designers. Though the sizes of the devices have shrunk, the design principles and portability solutions have changed amazingly little (Keinonen 2000). "*

According to Frank Nuovo, Nokia's chief designer, a mobile phone design "gives faces to a company, globally" (Helsingin Sanomat 2001). Nuovo compares a phone to *fashion statement*, such as watch or sunglasses. Nokia mobile phones are identified by the their continuity. The essential factors are big display, lack of external antenna, round shapes and ellipse around the display (Figure 4.1).

Figure 4.1. Nokia Mobile phone industrial designs. Nokia 5510 and Nokia 6210.

A good design has potential to create global identifiable *icons*. For example, Nokia 2110 design has been used globally as a symbol of a mobile phone in different contexts. In the same way, in some countries, for example China, the common noun for a mobile phone is "nokia". Technically, industrial design defines the overall dimensions of the product and the main factors for mechanical design.

4.2.2 Mechanical design

Mechanical product design is a detailed implementation of industrial design. Mechanical design initially defines the physical product implementation, such as materials, dimensions and positions of product components. Hence, industrial design together with mechanical design provides the key ergonomic design decisions. Mobile phone manufacturers have tried several mechanical basic concepts. Some concept variables are listed in the following:

- orientation: Horizontal vs. landscape product (Figure 4.1)
- covers: No cover, clamshell, flip, slide (keypad cover) (Figure 4.2)
- keypad: ITU-T phone keypad, QWERTY keypad, touch/pen keypad (Figure 4.2)
- display: different sizes, 1 vs. 2 displays (Nokia 9210), display colours/no colours
- softkeys (key label in the display, but physical button under the display): 1-4
- call management keys: Send/End keys vs. uni-key (Figure 4.4)
- navigation tool:
    - no navigation tool, 2-way scroll (Nokia 7110), 4-way scroll (Nokia 7650)
    - 2 navigation keys (up/down), 4 keys (up/down/left/right), joystick (analogue or digital), navi-wheel (Sony), roller (Nokia)
- internal vs. external antenna
- detachable parts: battery, SIM card, memory card, flip-on covers
- carrying concept: Hand-held, wearable (wrist phone, headset)
- need for hands in operation: no hands (vice control, one hand, two hands)

Examples of mobile phones industrial and mechanical designs from different manufacturers are presented in Figure 4.2.

Figure 4.2. Mechanical product designs. Nokia 9210 (clamshell), Samsung wristphone, Benefon Esc. Ericsson R380 (flip phone).

Innovative products are often presenting new ideas for user interaction, such as, new navigation tool concept, special display size or shape, experimental keyboard layout, moving components (flip covers, hinged camera), and built-in sensors. Each new interaction concept is a challenge for the product development (implementation risks) and user acceptance (usability, utility).

Industrial and mechanical designs form the physical interface between user and the phone. Underneath mechanics there is hardware that enables both mechanical and software functionality. The main mechanical components of a basic mobile phone user interface are displayed in Figure 4.3:

Figure 4.3. Mechanical components in a mobile phone user interface (Nokia 8310).

4.2.3 Hardware

Hardware defines main performance issues, such as display capabilities, battery consumption (together with SW), memory capacity and processor efficiency. Mobile phone hardware is highly and tightly integrated entity, typically without readiness for modifications such as extending or upgrading. In some phones it is possible to extend the memory capacity with special memory cards, for example Nokia 9210. The capability to optimise the size, compactness and functions is critical, due to continuous and competitive attempts to make the mobile phones smaller. Hardware enables certain software performance. Hence, software functionality is dependent on hardware.

4.2.4 Software

Telephone development started as hardware-focused engineering. However, the importance of software and SW based functions is continuously increasing. It is no longer possible to develop a mobile phone, especially a smart phone, without extensive software engineering. Mobile phone manufacturers have entered the software industry.

Practically, software *platform* defines what software functionality it is possible to implement and what is not with reasonable effort, and the software architecture defines how flexible the platform is, especially with complex products and rapidly changing technologies (Hofmeister et al. 2001, xxiii). For example, multitasking is not implementable with one SW platform, or activities requiring high processing capability will not run smoothly with another platform. Further, even if a software platform is ideal for multitasking and PDA tasks, it may be unsuitable or inefficient for cellular processing tasks, such as digital signal processing (DSP).

Mobile phone software is embedded software or system. This means that the software is integrated to other product components, such as specific hardware interfaces. Mobile phone software platforms can be grouped as follows:

- manufacturer specific (proprietary) platforms. These platforms are typically designed and evolved for the very specific needs of a company and they are not open for software developers. These platforms are optimised for and efficient in a specific hardware platform. These SW platforms require very specialised company-internal know-how and SW tools that are difficult to buy from outside.
- open software platforms. These platforms for mobile devices are developed for enabling open (3rd party) software development. Examples of these platforms are Symbian (www.symbian.com), Windows CE, Stinger (www.microsoft.com), and Geos (www.geoworks.com). These platforms may not be optimal for a specific company or hardware platform but they open better possibilities to outsource software development, know-how or find SW development tools. Symbian platform is an example of SW development where the actual SW engineering is done by partners, subcontractors, or even competitors.

Symbian (2001) lists principles of mobile phone software capability requirements. These principles may seem obvious but they are radically different to the current desktop applications:

- call handling and management of personal data must be possible anywhere anytime (user expectation)
- serious power management. The device needs to be responsive in all situations, and cannot afford to go through a boot sequence when it is turned on. It must always be able to raise alarms or handle incoming calls. It must provide many hours of operation on a single charge or set of batteries.
- at a very basic level the OS shouldn't be too resource hungry. It should support low-power CPUs with limited amounts of memory.
- at a deeper technical level, expensive operations such as context-switching should be minimised: it is better to implement most of the multitasking through event-driven messaging rather than with multithreading.

- code reuse should be maximised. This is a goal often stated but rarely achieved.
- the allocation and use of resources has to be tightly controlled. This can best be done on a system-wide basis using servers to control system resources.
- reliability is a major issue for mass-market devices. Data loss in a personal mobile device causes a loss of trust between the user and the device: a WID must be at least as resilient as paper diaries and agendas.
- the kernel should be small: much of the functionality conventionally handled by device drivers should be handled, instead, by system servers, running without special privilege.
- an effective memory management system is needed to prevent memory leaks. For systems that are never completely shut down and cannot be rebooted, keeping an accurate track of resources is what makes the difference between peak performance at all times and slow degradation to partial or total lack of usability.
- sound consumer design is necessary.
  - applications should take advantage of the uniqueness of the WID.
  - they should be designed both for current usability and for future developments in wireless technology.
  - consistency of style is paramount. If a function is too difficult to use, then it cannot justify either the time it took to develop or the space it takes in the device.

Software platform functions and UI style are often dependent on each other. The selection of software platform, for example Symbian platform, may define principles of the UI style for a product, or the selection of UI style may guide the company in selecting the software platform, for example Microsoft CE.

4.2.5 UI style

A *user interface style* is a design framework describing interaction style and objects, including appearance (look) and behaviour (feel) (Hix and Hartson 1993). In the world of desktop computers UI style refers to the software UI (display). For a mobile phone, user interface style defines both the display and the required keys and buttons in the device. In mobile world, there is no single ruling UI style as there is in the world of personal computers. Instead, mobile phone manufacturers are maintaining and developing brand specific UI styles. Even inside one UI style there can be hundreds of rules and guides for the UI design. For example, (in a non-published study) over 800 UI design rules and guidelines for PC environment were identified. In addition, some manufacturers have several different UI styles. Some examples of Nokia UI styles are shown below (Figure 4.4).

Figure 4.4. Nokia 7110 (left) has two softkeys and a "roller" for UI navigation. Call handling keys are green and red buttons. Nokia 3110 (right) has one softkey and up/down arrows for navigation. No special keys for call handling.

Mobile phone displays are small, with display area of few square centimetres. The evolution is driving to maximise the portion of screen size in the phone face, colorize the display, simultaneously decreasing the whole product size. Several phone manufacturers have already gone below the challenging 100g weight. In current products on the market (December 2001) the interaction is mainly based on physical keys and buttons instead of touch-input systems.

There are signs of emerging competition in using the UI style as competing issue, and pressures to provide a de-facto UI style in order to enable open software development for mobile phones. Symbian is providing UI platforms for different product categories and Microsoft is on the way to develop mobile phone user interface style (known as “Stinger[9]”).

Changing from one UI style to another is difficult for the user due to the challenge of learning new way to perform familiar tasks. Currently, the diversity of UI styles leads to the situation where the user has to learn new ways to perform tasks, i.e. adapt to the particular UI style when she replaces old product to a newer one. This is an important issue, because replacement customers are a major group in the consumer markets. During 2001 about 70% (Gartner 2001) of the mobile phones were sold to persons that already have one. At 2002 the replacement market of mobile phones is larger than the share of new customers.

4.3 A Mobile Phone Project

Mobile phone development projects followed in this study (Figure 1.2) can be characterised with the following attributes:

- projects are complex (organisation, product, user requirements)
- there is always too little time (and never extra time) for product development.
- there is always lack of resources
- timetables are always slipping
- plans and timetables are always optimistic
- some implementation risks come always true during the project.

These attributes seem to apply also universally in IT projects (for example, Langefors 1966, Keil and Mann 2000). Work will expand to fill the time allotted to it. Hence, they can be called "natural laws of IT projects". Cooper (1999, 43) presents the 90-90 rule for software development: “The first 90% of the code accounts for the first 90% of the development time. The remaining 10% of the code accounts for the other 90% of the development time”

Mobile phones are often launched to publicity in advance with estimations about the start of shipping. Very often the estimations are optimistic and shipping seems to be more or less delayed (from market perspective), and sometimes the product is never seen at the market place (Cooper 1999, 44). The problems are often consequences of difficult situations in the product development, infrastructure implementation or service availability, partly caused by the natural laws of projects, and the technical complexity of product, organisation or user requirements (Tianfield 2001).

4.4 Sequential development process

Sequential design process is the natural order for developing innovative products. The phases of sequential product design, for example the Waterfall method (Royce 1970), are requirement analysis, specification, design, implementation, integration and testing (Hakiel 1997). In the following the product development is studied as a sequential process. In the scope of this review

[9] “Pocket PC Phone”, www.microsoftmobilitypress.com

only upstream activities necessary to design and implement a product from development team perspective are discussed, i.e. downstream activities such as manufacturing and marketing are not discussed.

Järvinen (1999) gives a general model for sequential development where the development advances from the initial to the target state. In this model the product development has stages and the development advances linearly (Figure 4.5) to next target state.

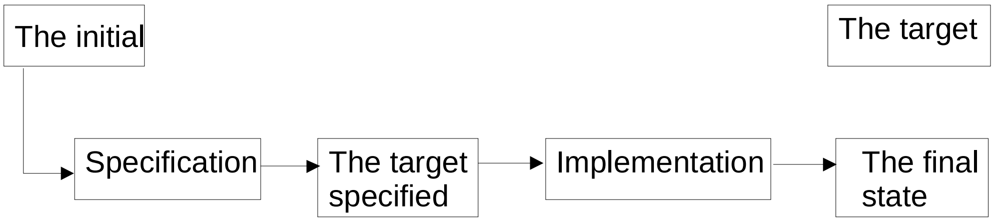

Figure 4.5. Building process for sequential design (Järvinen 1999)

The purpose of specification process is to consider various sub-methods to derive and determine the *target* state of the desired artefact under construction. Specification defines, *what* is the target. According to Järvinen, "the new artefact is intended to be used by users, and the engineers are assumed to best know the best idea. … Both parties have difficulties to understand each other, because their professional languages differ much."

In implementation process the organisation implements the specified artefact and tries to answer *how* the target should be implemented. The answer is sought, for example, by applying problem reduction heuristics, producing design alternatives, and building prototypes. Prototype and simulation development is also a parallel activity to the actual product development.

The sequential design process in this study has six *milestones* (M0…M5) and five development phases (specification, design, implementation and integration, testing and launch) (Table 4.3).

M0		**M1**		**M2**		**M3**		**M4**		**M5**
	Specify		Design		Implement and integrate		Test		Launch	

Table 4.3. Milestones and development phases.

According to Nunamaker et al. (1991) new artefact is always based on a concept. Hence, the Hakiel's definition of sequential process lacks an important aspect of mobile phone development: Concept design. Also, usability engineering lifecycle models of Nielsen (1993, 72-111) and Mayhew (1999) do not make distinction between product development and product concept design.

4.4.1 Concept design

> *Concept is an abstract idea, or an idea or invention to help sell or publicize a commodity (Oxford dictionary 1998).*

One or more ideas or innovations can be used to create a larger entity, a product concept. Product concept design is the early development phase that precedes the actual product development (King and Majchrzak 1997). It is the input data for product specification work. The output of concept design is a product concept or clear vision of the product. In concept design it is important to understand and precisely know how people act in the real environment using real devices and tools. It is not enough to ask, because a large portion of our daily tasks are automated or routinized and difficult to repeat by describing (Wikberg and Keinonen 2000).

> *Concept design is the adaptation and amalgamation of technological and theoretic advances into potentially practical applications (Nunamaker et al. 1991).*

A new concept is always based on an innovation in some resource type. The resource type can be L (material, technical), E (human, social), R (money), I (data, information, knowledge), or some combination of these. Most advances in information technology are based on technical innovations, but some innovations have been greatly helped also by social innovation, for example iMode, text messaging applications and chat.

Hamel and Prahalad studied (1994, 89) the ways how a company can build its future. They say that "to create the future, a company must first develop a visual and verbal representation of what the future could be." Product concept is a visualisation of the future in a smaller scope. It is the visualisation of what the product could be in order to make the product development team to clearly understand its goals. According to Frank Nuovo (Helsingin Sanomat 2001), many concepts never appear on the market place. Hence, there is a large amount of such early development and usability engineering that only builds the organisation's knowledge and memory, but is never seen in a real product.

In mobile phone development we need to separate product concept and UI concept. Product concept is the general description of a product, and it may or may not include UI concept. The main factors of phone design, core functionality and intended customers are included in the product concept. UI concept defines the main elements of user interaction, such as UI style, input and output techniques. Both concepting activities need considerable amount of research in several areas, for example, markets, technologies, users, trends, fashion, even sociology.

In the same way as there are many industrial design candidates for a product, there are often also several product concepts created for the product entity. The concepts are selected for product development based on information from many sources. One such source is the Customer Satisfaction (CS) process that studies what needs customers have and with what kind of products they can be filled (Valjus 1994). New technologies are constantly being explored for their applicability in new terminal concepts (Väänänen-Vainio-Mattila and Ruuska 1999). A specific method that is actively applied at Nokia for product concepting on function and UI level is Contextual design (Beyer and Holtzblatt 1998).

A mobile phone concept design can follow the process described by Ulrich and Eppinger (2000). The process has three main phases (User study, Concept creation phase and Concept evaluation) each with several detailed tasks (Figure 4.6). Social innovations (E) are sometimes sought by studying users or specific user groups in some context (User study). Mikkonen et al. (2001) give an example of concept design process in the context of mobile terminals and elderly users.

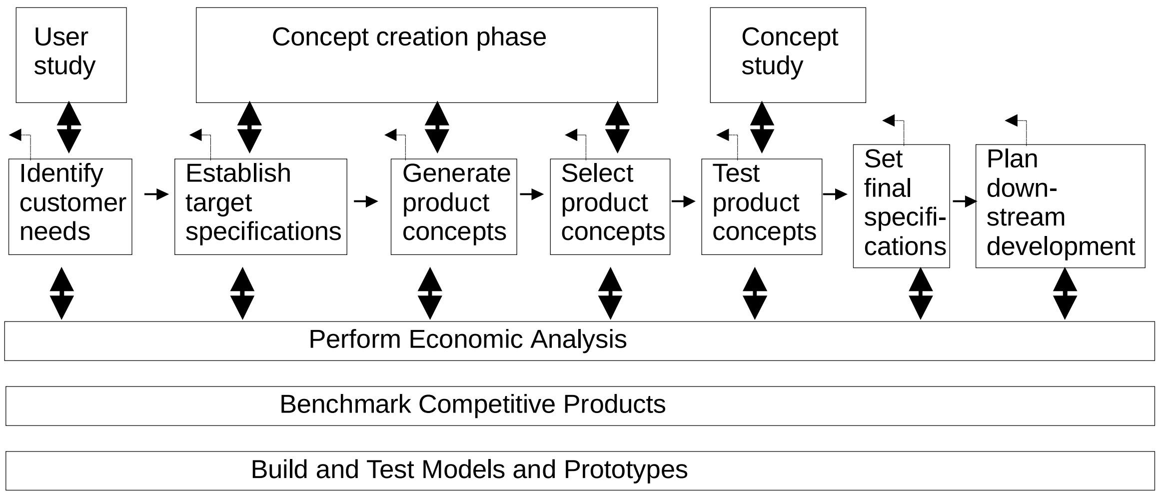

Figure 4.6. Needs identification, target specifications, and various conceptual phases in concept design (Ulrich and Eppinger 2000).

Concepting is important to the product development, because it is the main information source to start the actual product development with specification and requirements analysis, and to guide design decisions in the early development phases.

4.4.2 Specification and Requirement analysis

Requirement analysis is the first concrete step in building the product. Requirement analysis may be part of product specification but it can also be an external activity to a project. According to Truex et al. (1999) requirements are not necessary determined as part of a project, but become a negotiated outcome of the changing characteristics of an emergent organisation and the resources. They call the process *dynamic requirements negotiations*. Because the organisation is emerging around the users, requirements can never be fully specified because users are always in conflict with them. Critics challenge the assumption that requirements can be stated in advance once and for all by a “specification freeze”, and propose prototyping process model type which suggests that requirements and functions are developed in parallel (Lyytinen 1987).

When the organisation implements something new that has never existed, for example technology that enables a major social innovation, the organisation is creating a potential possibility or opportunity. In this case there may not be user needs towards the product, but the needs can be created, for example, by *marketing* activities when the product is launched. It can be difficult or even impossible to find out user needs by studying users when innovative products, functions or technologies are developed.

Product requirements are obtained from several sources, for example, the company roadmaps, competition information, and major customers. The requirements can originate from L-, E- or I-innovations inside the development team or from outside. In mobile phone industry, network operators are often the main customers having first-hand access to users because the phones are sold via the operators. Hence, operators can propose or set product requirements before they agree to sell the products.

Specification turns the diversity of product requirements to exact facts and targets to be achieved. Product specification defines what is the expected output of a development project. For a mobile phone, it can define, for example, size, weight, customers, network system (such as "GSM 1900"), time-to-market, and development cost/unit.

Requirements are not stable. User needs change (new needs emerge, existing needs change), and requirements that come from other external sources also change. On one hand, the development team needs to be able to respond to changing requirements. On the other hand, to change requirements during product development, especially in late phases, creates an implementation risk.

In this study, M1 is the stage where the organisation agrees about the product specification and requirements. When these are known, the organisation is ready to implement the product.

4.4.3 Design

Product design turns the specifications, requirements and product concept to product design and finally to product prototype. Product design for a mobile phone can be done in three areas: design of product (the user interface), design of product externals (the external interface) and service design (the service interface).

Product design focuses on the product itself. The main product design activities cover:

- finalisation of the industrial design (started in product concept creation)
- mechanical design
- hardware design
- software design
- UI design

Product externals design focus on designing elements that are not the product itself, but are part of the entity that is delivered to a customer. The main product externals design activities cover:

- user guide design
- package design
- accessories design

Mobile phones, especially innovative products, are core products of a mobile phone manufacturer. *Core products* are the physical embodiments of one or more of company's core competencies (Prahalad and Hamel 1990). Core products can also be re-usable components, such as software. Often, innovative smart phones introduce new service types, essential for the idea of the product. For example, multimedia messaging service[10] (MMS) is essential for products that create visual content. Services are core products of a service provider. Services can be based on E-innovations (need for a service), L-innovations (service is technically possible), or I-innovation (for example, service is based on a database that has not existed earlier or is possible to create now).

Often new phones do not have added-value compared to basic mobile phones if there are no appropriate network services available, and new services can not be used without appropriate terminal devices. For a successful launch of core products, the two (terminal and service) must meet in an affordable way. If this succeeds, it typically provides a "win-win-win" situation for mobile phone manufacturer, service provider and the user. Examples of this situation (network and service are needed for phone functionality, and phone is needed for the use of network and services) are

- introduction of new bearer services (for example GPRS):
- short message services
- WAP services,

[10] A messaging technology for transferring multimedia content between mobile terminals.

- multimedia services (MMS),
- location services and
- network games.

Service development needs to be part, or at least taken into account in the mobile phone development for successful implementation of core products. In mobile industry, mobile phone manufacturers and service providers are maintaining continuous dialogue for right-timed core product launches.

In this study, M2 is the stage where the organisation has design documents available for the whole product development. When design is completed it is time for implementation.

4.4.4 Implementation and Integration

Implementation turns the design documents to a concrete artefact, a product. In practice, the distinction of these phases is not strict, but implementation may have been started already during the earlier development phases when some parts of the product design are known even though the design entity is not frozen. In a CE project, product components are implemented in parallel. Integration is the activity that builds the product entity from parallelly implemented artefacts.

In this study, M3 is the stage where the organisation has implemented and integrated the product to a working product prototype with full functionality.

4.4.5 Testing

Testing is a coordination activity. It verifies that the specified and designed functionality is correctly implemented by finding deviations and monitoring the advances. The main activities in the testing are:

- module testing for smaller components, such as implementation artefacts
- system testing for larger entities, such as the product prototype
- field testing in market areas
- interoperability testing
- user acceptance testing

In this study, M4 is the stage where the organisation has tested the product and corrected the errors. The concrete output of this phase is well-working product. However, in the development of technically complex products, it seems that some errors always remain even after the testing, such as software bugs.

4.4.6 Launch

Product launch (M5) is done when the organisation is capable of manufacturing the product with required volumes. This is often the ending point for a product development project. Due to the remaining product errors, for example, in software, there are always activities that are dedicated for correcting those. In this study we call those Post-M5 activities.

4.4.7 Discussion on sequential process for innovative product development

The challenges in innovative product design are:

- specification is difficult and takes a long time due to limited knowledge about new emerging L-innovations that are implemented for the first time in the product

- specifications are not always fixed in the target specified state. In other words, specifications and even technical standards tend to change during the implementation.
- changing specifications produce delays for implementation.

Sequential development turns the larger problem (product development) to smaller manageable sub-problems (development phases). This is one way to apply problem reduction heuristics. This approach means that there is increased need for coordination activities, such as transfer of product components, inspections and communication in order to cope with the transitions between development phases (Järvinen 1980).

The sequential development process is problematic especially for UI design because the next target state is a moving target. When more or emerging information is available about user requirements, dead-ends emerge for implementation and new implementation possibilities are found, there are pressures to change specifications *during* the implementation, even when it is costly or risky for the project. Changes in technological level (SW, HW) can have impact on the user interface.

A simple example: at some point during the implementation of a new system, the development team may find out that the system performance (speed) is worse than was estimated or the system memory consumption exceeds all estimations (there will not be memory available for the user). It must be decided whether some functionality is dropped or changed in order to provide decent overall system behaviour. Whatever decision is made, it will have impact on user experience, usability and system specifications.

Technical functions are relatively easy to define and fix early, such as dimensions of the product. However, user interface issues are harder to define because they are often dependent on the information that is achieved from user testing and system implementation. It seems that parallel design strategy is more appropriate for SW and UI design than HW design. This observation proposes evolutionary approach where the target state definition and implementation are performed in parallel. As a conclusion, we can join the proposal of Järvinen (1999) that the target state can be achieved via two paths: through sequential building process or parallel building process (concurrent engineering) (Figure 4.6).

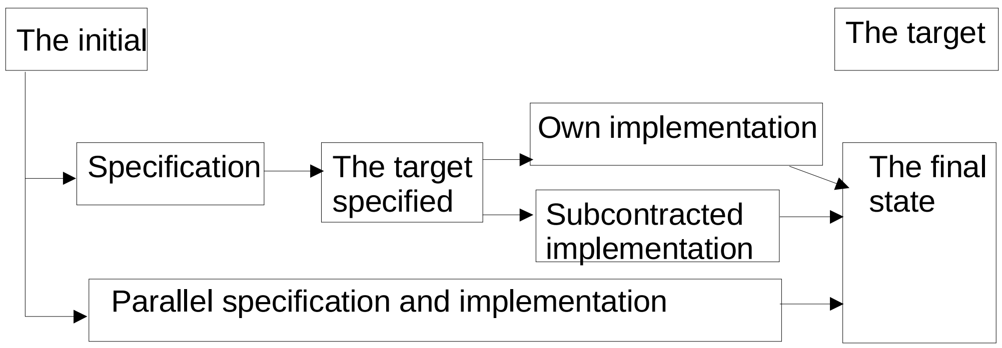

Figure 4.6. Sequential and parallel building process for achieving the target state Järvinen 1999).

4.5 Levels of complexity

Product development concurrency is a function of the number of parallel development processes. When product development concurrency increases, the complexity of product development increases. Evaluation of potential product development complexity can be based on evaluating the potential maximum rate of development concurrency:

- 1 sequential process = complexity 1 (dependent on one other process, i.e. itself)
- 2 sequential processes (A&B) that have dependencies = complexity 2 (A is dependent of B, B is dependent of A)
- 3 sequential processes (A,B,C): A-B, B-A, A-C, C-A, B-C, C-B) = complexity 6

In the worst case the complexity of product development exponentially increases when the number of concurrent processes increases. In addition, the complexity is also increased by increased need for coordination tasks.

Concurrent development process can be based on two different approaches: either the sequential development process is compressed to overlapping phases (Figure 3.3), or there are several parallel development processes (Figure 3.1). The most challenging form of concurrent product development is then the scenario where there are (a) many parallel (b) overlapping sequential development processes having dependencies between the processes (Figure 4.7).

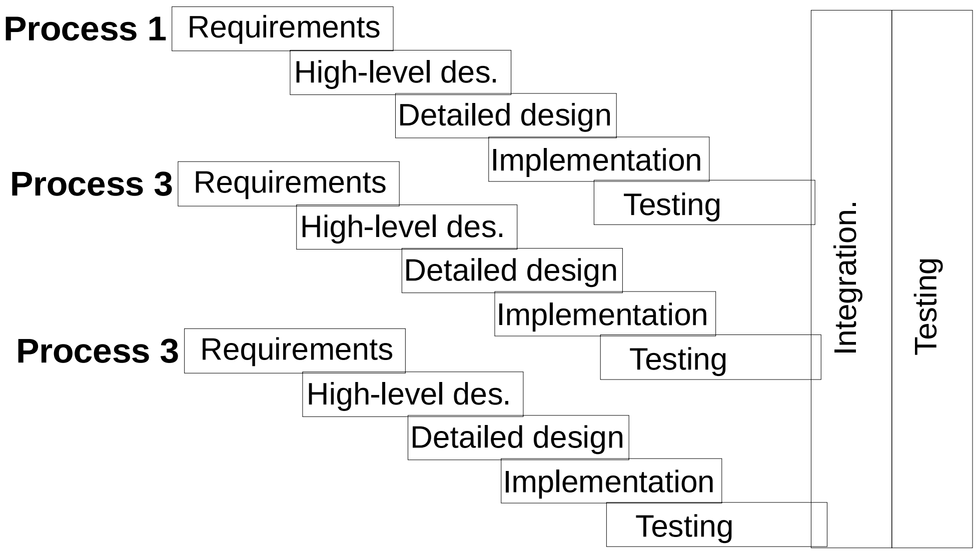

Figure 4.7. The most challenging scenario of concurrent product development.

In concurrent product development, product requirements need to be further subdivided to process requirements. For example, product requirements for implementing GPRS functions need to be stated as SW requirements for SW engineering, HW requirements for HW engineering, UI requirements for user interface design, etc.

The overlapping phases were seen in all main processes (for example SW, HW, mechanics, UI, user documentation, packaging, localisation) during the whole product development lifecycle. The milestones were not exact turning points from one design phase to another (previous phase ends and new phase starts), but in each milestone there were two ongoing phases (for example detailed design and implementation) in each process. In addition to overlapping development phases, interdependencies were continuously observed between development processes. Some parallel development processes that had noticeable interdependencies were:

- Product concepting - Industrial design
- Product concepting - SW design
- Product concepting - Mechanical design
- Industrial design - Mechanical design
- Industrial design - graphical design (colors, graphics)
- Mechanical design - HW design
- SW design - UI design
- Localisation - UI design
- UI design - User documentation design
- HW design - SW design

4.6 From tunnel vision to shared understanding

Design and implementation of a mobile phone requires capabilities to systematically create and implement an appealing product from given design elements, applying the concurrent development process. Innovative smart phone development is challenging due to uncertainties related to new technologies and the high rate of concurrency in implementing the technologies in a user-friendly way.

The product development process can be conceptually analysed and phases labelled, but in practice it is difficult for an engineer to perceive and identify the real state of the product development. "Is the UI design now frozen, or will there be more changes due to software changes?", or “is it still possible to change design X?”

The special challenge in concurrent product development is to enable real-time analysis of the whole product. Oxnevad (2001) proposes that concurrent design process "should be seen as continuous process rather than as a sequence of phases with high walls between them." To overcome the difficulties in product development, three initiatives can be proposed: concurrent design sessions, real-time analysis with high-fidelity prototyping tools and shared product vision. In concurrent design sessions (Oxnevad 2001) the members of different design teams, forming hence multi-disciplinary teams, work together in sessions where current design and implementation is analysed using high-end analysis tools, in near real-time fashion. This ensures that a total systems approach is taken, and that all relevant engineering and sciences are covered. Mobile phones are technically well-defined devices that can be simulated and prototyped. Real-time analysis, including user testing, of the product can be performed using a high-fidelity prototype whenever needed and when there are resources available to implement sufficient prototype. The noticeable observation is that only about 5 years ago, providing high-fidelity and high-end tools for real-time analysis was impossible (Oxnevad 2001).

According to Parker (2001) even experienced discipline engineers have limited view and expertise in other disciplines, outside their own expertise. The more novice an engineer is, the more limited is her/his view. In the complex product development, one-discipline tunnel-vision is a considerable problem, especially when one-discipline engineers are elevated to lead system engineers or maybe program/project managers. The effect of tunnel-vision is that a discipline engineer is unaware of the traps waiting when knowledge from other disciplines is required. Parker proposes that all new discipline engineers should be given training at the beginning of their careers that will make them aware of all engineering disciplines in the company, how they generally depend on each other, and how experienced experts go about doing their jobs. The ideas of Parker can be further applied in product development. However, the proposal covers only the area of L-innovations (material, technical innovations), but does not cover other important areas, such as E-, I-, and R-innovations.

> *Discipline engineers should be given training at the beginning of each product development project that will make them aware of the product concept in detailed level, of all engineering disciplines in the project, how they depend on each other, what trades and analyses must be done, lessons already learned, and how experienced experts go about doing their jobs (Parker 2001).*

The key for solving the challenge of discipline engineering and tunnel vision in innovative and complex product development is effective and continuous communication throughout the product development process. A potential and natural development opportunity is found by combining the idea of multi-disciplinary CE teams (Winner et al. 1988, Oxnevad 2001) and the need for multi-disciplinary work in UI and usability work (Norman 1998). The importance of knowledge production, not only inside a community, but also between communities are considered to be critical for the design of complex systems (Brown and Duguid 1991, Boland and Tenkasi 1995).

It is possible to identify the key persons in the projects of this study with good understanding about the product entity and expected output:

- product managers have the clearest view on the product entity and market requirements (from technical and project perspectives).
- persons who are creating user documentation (manuals) have good understanding of the product entity and all the product details that are visible to a user.
- usability engineers know the product throughout and have best on-line view on the problems that users will potentially have with the product.

These roles in a project are the most potential and competent sources for energizing knowledge sharing and communication about cross-disciplinary issues, as well as making team members aware of the product concept in detailed level.

4.7 Summary

In this chapter I have described the main characteristics, project phases and design areas in a mobile phone development project. The main finding is the fact that projects have a lot of common problems that lead to design uncertainty, such as increasing product complexity, demanding timetables, resource gaps and late change proposals.

5 Mobile Phone Usability

Our definition of usability, as given in Chapter 1, is a fundamental concept that remains the same for any interactive system. The same characteristic applies also to other definitions of usability - they are domain independent. Too difficult-to-use functions cannot justify either the time it takes to develop them or the space they take in the device. Naturally, the *factors* that define usability of an artefact are highly dependent of the artefact. In this chapter I discuss factors that affect the usability in a mobile phone.

For a long time, landline phones were mechanical devices without software. Also the first mobile phones were mechanical radio devices without embedded computers. The two fundamental differences between landline (and early mobile phones) and cellular mobile phones were, already in the beginning, the mobility (no wires) of the phone and software based user interface functionality of mobile phones. Conceptually, landline phones were communication centres of a place, such as home, and mobile phones became communication centres of a person (Kasesniemi and Rautiainen 2001, 104).

First generation mobile phones were designed to satisfy the same *user need* as landline phones - to make and to receive phone calls. The functions were designed to enable basic telephony. Later on, user needs have evolved, and the importance of software has been continuously increasing. The fundamental need, personal communication, has remained, but it has taken new forms. New, non-telephony functions for personal use have been embedded to software, such as text messaging, calendar, and at the end of 1990's, Internet connectivity. Also the set of phone accessories, for example chargers, replaceable covers and headsets, has widened. Even though the use of mobile phone may be fluent and familiar, not many users are able to define some essential elements of a mobile phone, such as the SIM card (subscriber identity module) or the role of cellular network or mobile phone operator (Kasesniemi and Rautiainen 2001, 76).

Simple phones are evolving to the direction of multi-purpose smart phones enabling rich communication between people, management of personal data and functions for entertainment. In order to be able to use the phone effectively, the user needs to deal, in addition to function-rich phone user interface, with external devices, diverse networks and services. A connection to person or service costs money to the user – an issue too often omitted in the product design.

From design point of view, a mobile phone is, first of all, an *interactive system*. ISO 13407 defines an interactive system as "a combination of hardware and software components that receive input from and communicate output to a human user in order to support his or her performance or a task". From user point of view, a mobile phone is an *information appliance*: "An information appliance is designed to perform a specific activity, such as music, photography, or writing. A distinguishing function of information appliance is the ability to share information" (Bergman, 2000). Finally, a mobile phone is a personal *communication system* enabling communication between humans and also between human and another interactive system (Figure 5.1).

The easiness of performing different tasks is dependent on the combination of product design, user characteristics, context of use and the task. In order to understand the usability of a mobile phone we need to understand these factors.

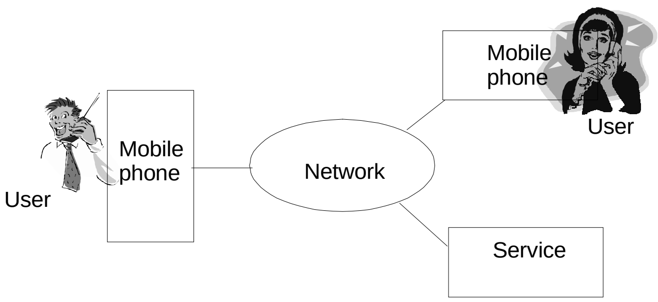

Figure 5.1. Phone, network and service

5.1 Product design

A mobile phone is a personal communication device (function), which has also aesthetic appeal (design). Recent studies have shown that a mobile phone can even be part of a person's identity with an emotional relationship (Kasesniemi and Rautiainen 2001, 191-192). Keinonen (2000b, 95) considers function, appeal and emotions, and presents usability related criteria that are present when the user evaluates and selects an electronic product: functionality, logic, presentation, documentation, usefulness, ease-of-use and affect. As long as business markets are more important than consumer markets, function reigns over design. As mobile phones become consumer products and closer to consumer markets, form begins to reign over function (Steinbock 2001, 268).

Consumers perceive concrete product attributes, and form beliefs that have influence on their attitudes. Concrete beliefs have an effect on higher-level constructs, such as benefits and values. Keinonen (1997) studied *expected usability*, which refers to subjective assessment concerning the quality of a product prior to actual practical experience. Expected usability has major impact on user behaviour, especially when novice users are concerned. It is based on user's cognitive beliefs and personal values (Aulin 1982). Keinonen claims that none of the usability dimensions (ease-of-use, presentation, logic) alone has remarkable influence on product preference, and they are incompetent in explaining consumer's decision criteria.

Usability, including aesthetic appeal, is still an obvious goal of product design. The importance of usability has been proved several times, also from economic point of view (Harrison et al. 1994). Nevertheless, usability is sometimes sacrificed in product design because of compromises with development timetable, function richness, tempting industrial design (appearance), low price, reliable technique, etc. (Keinonen 2000, 93).

Digital convergence introduces new usability requirements when technologies are embedded to everyday appliances: "Just like electricity, this new phenomenon is an enabler for us to do other things, many of which we may not even be able to imagine today, but which may become vitally important to us in the not too distant future. We are moving from a technology-centric to user-centric telecommunications world" (Nokia, 1999). Along the convergence, innovative functions are invented. Sometimes the functions are not responding to user's earlier experience or user needs, but it is up to user to find the value of the new functions.

Shneiderman (2000) refers to the same phenomena but from different perspective as requirement for *universal usability*: "Universal usability can be defined as having more than 90% of all households as successful users of information…The complexity emerges, in part, from high degree of interactivity that is necessary for information exploration, commercial applications, and creative activities." High degree of interactivity is a fundamental characteristic of a mobile phone. Hence, I propose extending Tianfield's (2000) list of product development complexities. The product development complexity results from:

- complexity of the product's structure,
- complexity of development organisation,
- complexity of user requirements, and
- *complexity due to convergence of different technologies*

Mobile phones of today are universal and global consumer products, and design reigns over function. There are two major trends effecting to product design:

1. Product variety (technical complexity and number of functions) is constantly increasing. A new product is never simpler than previous products. This is also seen as increasingly complex product development organisation, because new functions are often designed and implemented by new people or through new job design (The Law of Requisite Hierarchy).
2. Variety of the groups of users is increasing because mobile phones are available to increasing number of users in different cultures.

The number of functions increases by three ways:

- new functions are copied and embedded from other appliances to the new product
- new functions are invented and embedded to the new product
- functions are inherited from previous products. For example, WAP browser was first on high-end phones (Nokia 7110), but it came later to low-end phones (Nokia 3330).

Identifying and managing user requirements are increasingly difficult tasks because the number of offered functions in the interface is increasing, while the interfaces remain limited. In the following sections, I shall first discuss the mobile phone interfaces, and then the user.

5.2 *Mobile phone interfaces*

A mobile phone user interface is an integrated entity that is built from several hardware and software interaction components (example in Figure 5.2). These technical factors and details define the familiarity, look and feel, and usability of a mobile phone.

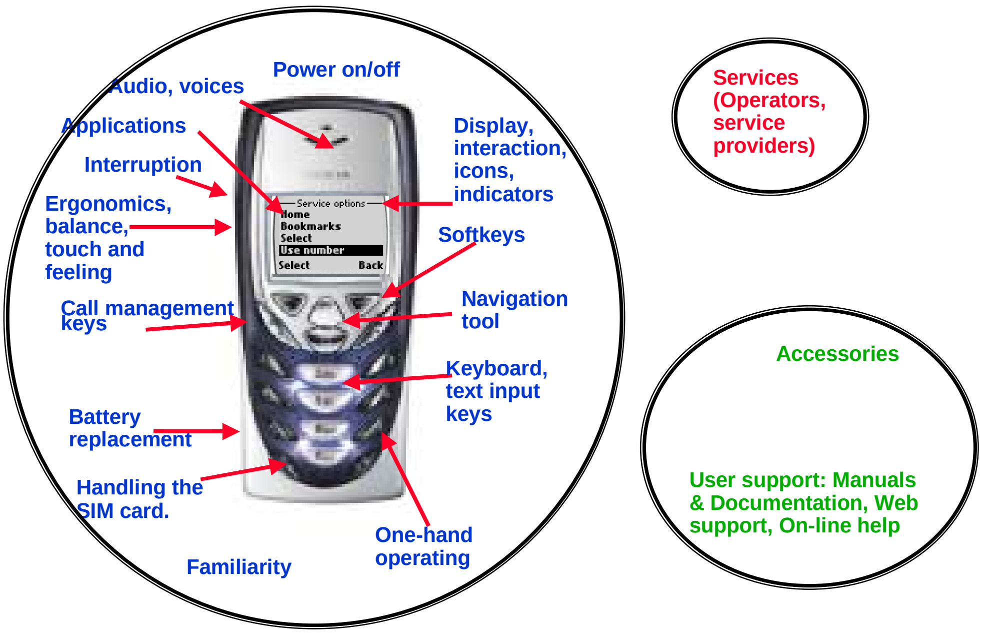

Figure 5.2. Factors affecting to usability in a mobile phone.

5.2.1 Mobile phone user interface and external interface

We can divide these interaction components to the three different categories: User Interface, External Interface and Service Interface (Table 5.1).

The User Interface category includes input and output devices and techniques, industrial and mechanical design and application (software) factors. User support elements, accessories, PC connectivity and downloadable software form the External Interface of a mobile phone. External interface is the interface that helps to use the device but is not physically part of it. Though it is important part of the overall usability, it is too often omitted in usability engineering. Service Interface is provided by the wireless network, as described in Section 5.2.2.

Interface	Category	Components
1. User interface	**Input tools (functionality, industrial and mechanical design)**	Navigation tool, Softkeys, Keypad/ keyboard, Special keys (Power, Call management, Voice)
	Display	Icons, Indicators, Language,. Familiarity, Localisation
	Audio, Voices	Ringing tones, Quality, Interruption
	Ergonomics	Touch and feeling . Slide, one-hand operating. Balance, Weight, Size
	Detachable parts	SIM card, Battery, Snap-on cover
	Communication method	Radio link, Bluetooth, Infrared, Cable
	Applications	Fun, Utility, Usability
2. External interface	**User Support**	Local help, Manuals, Documentation
	Accessories	Charger. Hands-free sets, Loopset, External keyboard
	Supporting software	PC software, Downloadable applications
3. Service Interface	**Services**	Availability, Utility, Interoperability

Table 5.1. Interaction elements and factors that affect to mobile phone product usability

Input tool of a mobile phone is typically a keypad. Sometimes voice input is possible for phone operation. Voice is the primary information and communication channel in basic phones. Data communication is increasingly important channel for smart phones and wireless information devices. Mobile phone interface is operated with special control tools, such as navigation and call management keys. Output is given as audio (speaker) and as display activity (icons, indicators). The industrial and mechanical design define whether the phone is pocketable, user can use it with one hand and has a nice touch, if the SIM card and battery handling are easy, and if the buttons are easy to press.

Mobile phone manufacturers are developing and maintaining brand specific UI styles. This means, that usability is also affected by such factors as, for example, familiarity and consistency of terminology between products.

5.2.2 Mobile phone Service Interface

In order to understand some characteristics of mobile phone use, we need to separate user interface from service. *Service interface* is users' view of an available operator or service provider's service through the phone UI. Service interface enables access to existing cellular infrastructure, such as basic voice communication, and services provided through the infrastructure, such as the Internet. For example, WAP browser is part of the user interface, but a WAP service belongs to the service interface. Due to mobile phone interaction and UI design conventions it is sometimes difficult and confusing for the user to understand which part of the service is phone functionality and which belongs to the service.

The next figure (5.3) provides an example. In Internet browsers, such as Netscape, there is always user-defined function "Home" available in the menu (Figure 5.4). "Home" takes the user always to the same place making it context independent. There is similar function in some WAP browsers, for example in Nokia 7110 where the exact Internet address (URL) of "Home" item in Service options list depends on the currently used service, i.e. it is context dependent. Other options (Bookmarks, Select, Use number) are provided by the phone UI and they have always the same meaning, i.e. they are context independent. In addition to that, due to the underlying WML protocol, the service provider can define whether Home option is available at all. The users are not able to guess or know where in the user interface or in the service they are taken when option "Home" is selected.

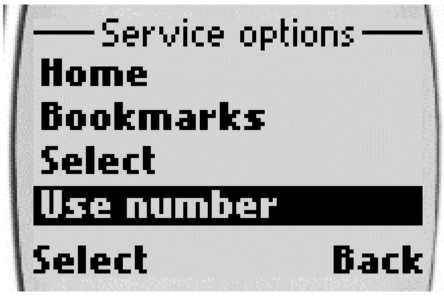

Figure 5.3. Mobile phone as service interface (Nokia 7110). Notice the double "Select".

Figure 5.4. "Home" in Internet browser.

In cellular networks, services have initially been voice communication, voice mailbox, text messaging and simple data services provided by the network operators. Cellular networks and terminals are evolving to the direction where IP (Internet protocol) networks and cellular networks are converging, and the same services are available in both environments. For example, banking services are available in Internet with WWW browsers, and the same services can be used with a mobile phone either with voice interface or via WAP browser. Convergence will also create new services that are based, for example, on multimedia messaging, positioning technologies, or remote control capabilities. The key factors in the service interface are cost, quality, availability, utility and interoperability of the services.

Mobile services are not yet mature. Many current (WAP and text messaging) services are difficult or slow to use, or to take into use. Services are not usable. This was reported by Finnish consumer service department (Kuluttajavirasto 2001) in a market review about 491 mobile services. A special problem is that consumers are often billed even in the case when the connection fails or is interrupted.

5.2.3 Structuring mobile phone interfaces

The three mobile phone interfaces can be organised to a hierarchical system (Figure 5.5).

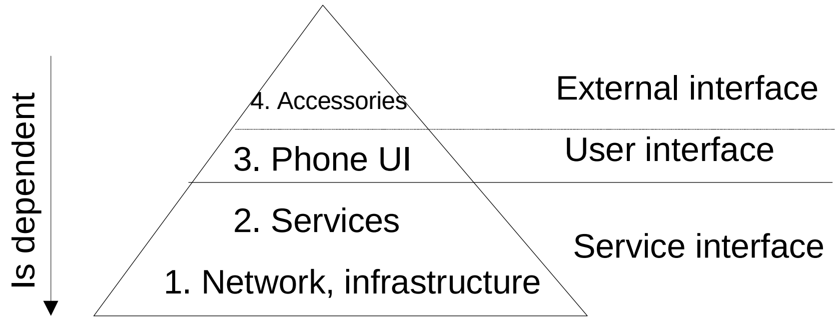

Figure 5.5. Interface hierarchy

Each interface in Figure 5.5 is dependent on underlying layers. For example, Services can't be accessed without network structure, Phone UI is not useful without Services, and Accessories are not functioning without a phone. In several usability tests I have observed that user problems are most often related to the highest level in the hierarchy. For example, in normal phone use the problems in network connection or failures in service use are often blamed to be phone UI issues, or if user has problems with headset, the specific accessory is blamed for problems.

5.2.4 Operators and service providers

The user experiences a usability of a mobile phone in specific context of use by *using the phone through* an available *operator service* (cellular network) in order to connect a *service* (or person) that provides the *content*. The experienced usability is an outcome of these all.

The *network operator,* that provides the basic connectivity, forms the communication platform for a mobile phone. Depending on the network quality and accessible services the user is able to perform his tasks. There are situations when the user is not able to use his phone temporarily or permanently. To manage network dependent problems the phone needs functionality to inform the user about the status of the network and about the available services. In some cases, a service may not be automatically available for the user. For example, voice mail is a network function that typically needs to be activated or purchased in addition to the basic service.

A *service provider* provides a specific service for the user. For example, Internet access is sometimes provided by some other organisation than the network operator. Especially in the case of Internet based services, such as WAP and email, Internet access may require configuration in the phone as well as agreement with the service provider (in order to be able to bill the user).

5.2.5 Feedback and feedthrough

Mobile phones enable communication between user and different types of terminals via cellular networks. The physical distance between the two ends can't be predicted or seen in the phone user interface. The communication quality is dependent on the prevailing capabilities of terminals and network systems. Dix et al. (2000) noted that *feedback* (seeing the results of one's own actions) and *feedthrough* (seeing the effect of other user's actions) are essential for interactive mobile systems. Communication delays and failed connections are more usual than in wired communication.

Feedback is a user interface issue that often can be analyzed and improved as part of product development. Feedthrough is a service interface issue that can't be fully analyzed or improved as part of product design, because it is dependent on the network latency and bandwidth. Mobility brings yet more issues, such as small screens and restricted input devices. Slow feedback and feedthrough problems are familiar for mobile phone users. Improving feedback and feedthrough related problems is an identified and strong driver in the development of mobile terminals and cellular network technologies.

5.3 Effects of good and bad usability

If the usability of a mobile phone is poor, it has effect on the user, operator, service provider and manufacturer performance. From a user's point of view, the user is dissatisfied with the product or with some phone functionality. The user may be even incapable of completing some tasks or to use a service. Unfortunately, users can not always distinguish whether some technical or usability problem is caused by the phone UI, network or by the used service. From the operator and service provider point of view, poor usability leads to decreased use of services and networks. For a mobile phone manufacturer, poor usability is seen in sales numbers, customer support services (Help lines) and in field failure rates.

An example of a problematic area is the *configuration* of a mobile phone for WAP connection. In the simplest situation a device can be configured by an operation that is provided by a service provider (Example 1). In the worst case, a user can't set up his device at all (Example 2). In this case, the usability is not built only by designing a good user interface, but it needs support elements outside the device. Example 2 is a good example of insufficient out-of-box readiness (OBR).

> ***Example 1:*** *"WAP phones Hyperactive. Nokia spares its customers the often tedious manual setup process needed when registering a new WAP service. When user's click on a web-site's Nokia Activ button and type in their mobile number, the settings for the service are sent as SMS message that automatically configures the phone." (Time 2000)*

> **Example 2:** *Nokia's Communicator Needs Work Before Consumers Try to Go Online*
>
> *“Here is my conclusion: With the 9110, I've found perhaps the most obtuse, infuriating and pernicious high-tech device that I've ever stumbled across. The instructions didn't help a bit. The 171-page owner's guide, the 30-page "Quick" Guide, and the help function on the accompanying CD-ROM read like parodies of user manuals. So after having devoted chunks of several days to the effort, I couldn't figure out even where to begin to set it up for Internet use. “ (Wall Street Journal, October 5, 1999)*

5.4 First use and out-of-box readiness

Mobile phones and information products are sometimes difficult to take into use because of several operations that are required by the user prior to use. In a mobile phone these are, for example, inserting SIM card, charging the device, configuration (settings) and personalization (ringing tones, replaceable covers, downloadable operator logos, etc). Several usability tests and general user comments clearly indicate that configuring a mobile phone for Internet connectivity is one of the most difficult tasks that the user faces. Internet connectivity challenges are seen in the use of personal computers as well.

While IT consultants are typically involved in helping consumer organisation with installation of new software and incorporation of that software into its existing information systems (Sawyer 2001), such assistance is not often available for mobile phone users. This problem is also addressed in previous Example 2, and one solution to the problem of configuration is given in Example 1 through automating the challenging configuration task. Due to the increasing product diversity, extending operator services and service provider functions, mobile phones have increasing number of settings, mostly related to Internet and messaging functionality, that are required for proper operation but cannot be provided as default settings. From product development perspective, this problem is addressed as out-of-box readiness (OBR).

Out-of-box readiness is a design goal describing the situation how easy or difficult it is for a user to start using a new product. IBM (2001), considering OBR to be the most important factor in project success, presents the motivation for out-of-box experience (OOBE) through potential impacts:

- the highest percentage of calls to help desks typically occurs shortly after product purchase. At this point the user is unlikely to be familiar with the product and may experience a variety of problems from unpacking through setup and configuration. Thoughtful support during this phase of product-user introduction can ease the experience and create a lasting, respectful bond between product, manufacturer, and user.
- this is also the point at which a user who made the decision to purchase the product should reaffirm that their decision was a wise one. Whether the user who unpacks and sets up the product selected it or not, their initial experiences set their expectations for future interaction and use.
- the out-of-box experience may also be the user's first direct interaction with the company and the brand. The user's image of both company and brand may be improved by a positive experience, or be damaged possibly beyond repair by a negative one.
- sales of products through retail and catalogue order channels are affected by how well the product presents itself to users; how well it communicates its functions; and how well it distinguishes itself from its competition. Packaging is an aspect of product design that contributes both to advertising and to the user's initial experience.

According to IBM, "Specific attention to the design of the OOBE should result in reduced support costs, improved customer satisfaction, and increased sales." In OOBE the problem lies very often in the required learning times with a new appliance. Sinkkonen (2001) studied the first use of a cordless phone with the research question how devices or services should be designed for novice or casual use. The general observation is that surprisingly few studies are published about this area.

OBR is a design goal of many parallel design teams and product coordination. It requires co-operation of development teams: product designers, product packaging designers, user manual designers, product support services, and even phone operators. For example, cellular operators provide several methods for easy phone configuration through smart messaging, such as one-click sending of required WAP settings over the air. A typical user problem arises when the phone manufacturer uses different terms than the operator, for example in the user interface and manuals.

5.5 Usability in some functions

In this Section I review some basic ideas specific to mobile phone use. A mobile phone needs, or is expected, to *informate* (Zuboff 1988) and tell the user about system activities, such as missed and unanswered calls. Informating occurs as processes, objects, behaviours and events are translated and made visible as explicit. A mobile phone is usable if the user can manage phone calls in the context regardless of other activities, such as walking. This leads to situations where the user cannot or does not want to receive phone calls, and handling of activity *interruption* must be supported.

A phone call consists of UI activities, simultaneous audio input and output, network functionality and recipient status (answer, no answer, not reachable). The call making sequence is usable if the user can enter or search the phone number from contacts list easily, see the status and availability of network and establish the phone call with sufficient audio quality. The finding of contact information and call establishment are phone UI issues that require use of the physical keys of the phone as well as clear feedback and information on the display. The network dependent issues, such as audio quality, are not phone UI dependent but they are easily perceived as they were.

Call management, as any communication activity in a mobile context, is technically a complex entity that can succeed or fail. Failure in communication is a common situation caused often, not by the device, but by the network, service or recipient. Handling of failures in communication is a usability issue that needs to be solved in the product design. The user experiences satisfaction if the communication succeeds, but is unsatisfied with *his phone* if the communication fails, for example, because of poor network coverage. The experienced usability is related more often to the phone than to the service or network.

5.6 Users and customers

In this Section our focus is in a customer and user. An essential part in the product development and in usability engineering is to understand, who are the customers, and hence who should be involved in the usability testing activities as test persons.

> *Today, everyone is a potential mobile phone customer. As the market has become increasingly segmented, the ability to master various product categories has become crucially important. In a segmented consumer market with high volumes, critical success factors include comprehensive product portfolio, a strong and appealing brand as well as efficient global logistics (Nokia 1998)*

Different user categorizations are being used for different purposes. The three dominant ways to group users are:

1. Categorization based on expertise: novice, casual, and expert users.
2. Categorization based on product buying or adopting behaviour: Early and late adopters

3. Categorization based on segmentation, especially lifestyle segmentation

Anybody can be a mobile phone user, independently of the age, sex, culture, physical disabilities, economic background, consumer behaviour or any other identified factor. This is a challenge for usability engineering, because one of the main activities in the early development is to know the user (for example, Nielsen 1993). How can we know or satisfy anybody? Design-for-all seems to be more useful guideline than design for specific users. Nevertheless, Cooper (1999, 124) claims that better results can be achieved by designing for identified single persons instead of "anybody" or large audience.

When the users are selected to participate in product development as test persons, it should be considered what user characteristics potentially provide useful information in the specific situation. Most often user testing is performed with users that are novice with the new system but may have previous experience of the task. Hence, usability tests tend to test *instant usability*[11] of the system. During the test sessions users learn and become, more or less, casual users of the system. However, expert users are rare. Along the development of innovative functions, the real challenge in human-centred design is to find expert users for product testing.

A system can be instantly usable or it may require learning. Grossman et al. (1992) propose the following categories for system learning:
- intuitive. The system is immediately usable, walk-up-and-use.
- discoverable. User can learn to use the system without need for external help, such as manuals.
- learnable. Manual or training is needed before the system can be used.

5.6.1 Novice, Casual, and Expert user

The most applied user categorization defines *novice*, *casual* and *expert* users (Nielsen, 1994) which predicts learning. This categorization describes user's experience with the specific user interface, knowledge of the task domain, and experiences about computers in general. A novice user has no (or only minimal) experience with the task domain and the system. A casual user is a person who is using the system intermittently rather than having the frequent use that an expert user has. The three dimensions can be organised as in Figure 5.6.

Nielsen (p. 28) shows that learning curves are different for novice and expert users. A system that focuses on the novice user is typically easy to learn but less efficient to use. A system that focuses on expert users is hard to learn but more efficient to use. The learning curve can be criticized with two arguments:
1. The learning process from novice to expert is not linear activity, but rather curvilineal activity.
2. Experience and knowledge are not synonyms. Experience and knowledge do not guarantee that a user is an expert user in a specific task.

[11] Also known as "walk-up-and-use usability" in HCI.

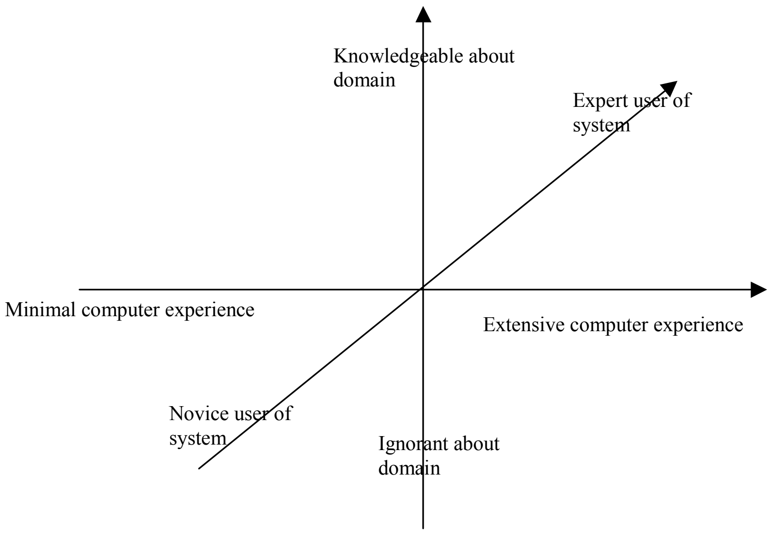

Figure 5.6. The three experience dimensions (computer experience, system experience, task domain expertice) (Nielsen, 1994).

For mobile phone interaction, I define a novice user to be a person who hasn't used a mobile phone or has limited experience with it. For example, a novice user may not have made or received a phone call with a mobile phone. A casual mobile phone user may own a mobile phone, using occasionally some basic functions of the phone, for example the built-in Phonebook. His lifestyle is not based on the use of a mobile phone. An expert user has his mobile phone always with him and he uses different functions fluently and often. An expert user has previous experience of mobile phones.

The larger factors that are forming the novice-expert categorization are:
- knowledge about mobile communication (context)
- experience about mobile communication (time)
- personal use of mobile phone (use frequency)

Nielsen's experience dimensions can be applied for mobile phone use by having the mobile phone as system, and mobile communication as domain and state of knowledge (Figure 5.7).

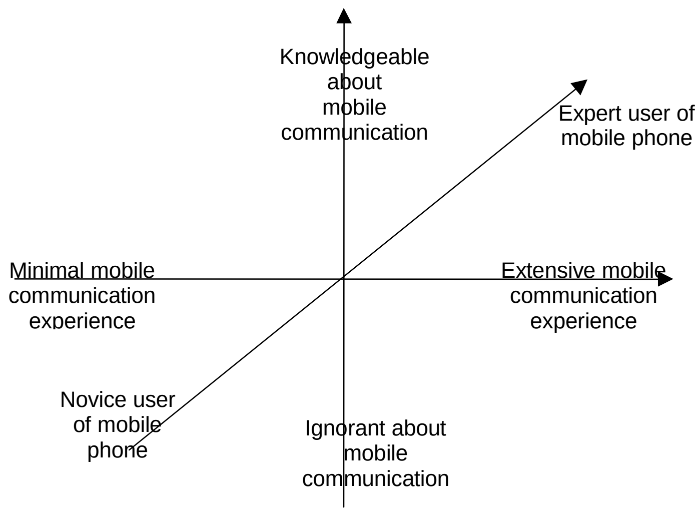

Figure 5.7. The three experience dimensions for mobile phone use.

5.6.2 Early adopters and late adopters

Norman (1998, 31-36) reviews market adaptation for a product through technology centred customers and consumer-centred customers, also known as early adopters and late adopters. This categorization can be used to predict buying behaviour and technology adoption.

In the early days of technology, some people will buy a product because of the functions it offers. These customers are called early adopters. The buying decision is primarily based on the function lists and technological claims on advertisements. Early adopters are willing to suffer inconvenience and high cost to get the technology. Late adopters, more conservative customers, wait until the technology is reliable, cheaper, convenient, and provides better performance.

Rogers (1995) discussed the extended concept of early and later adopters. He presents the categorization of adopters: innovators, early adopters, early majority, late majority, and laggards, each playing a different role in the development of technology. He showed that innovation spread slowly, with early adopters being different kinds of people than late adopters. Norman proposes that it is not enough to change the marketing for different adopters, but the entire product must change (Norman 1998, 274). In the early days of a technology or product, the innovators and technology enthusiasts drive the market. In the later days, the pragmatists and conservatives dominate with the need for convenience and solutions. The innovators and early adopters are only a small percentage of the market. The big market is with the pragmatists and the conservatives (Figure 5.8).

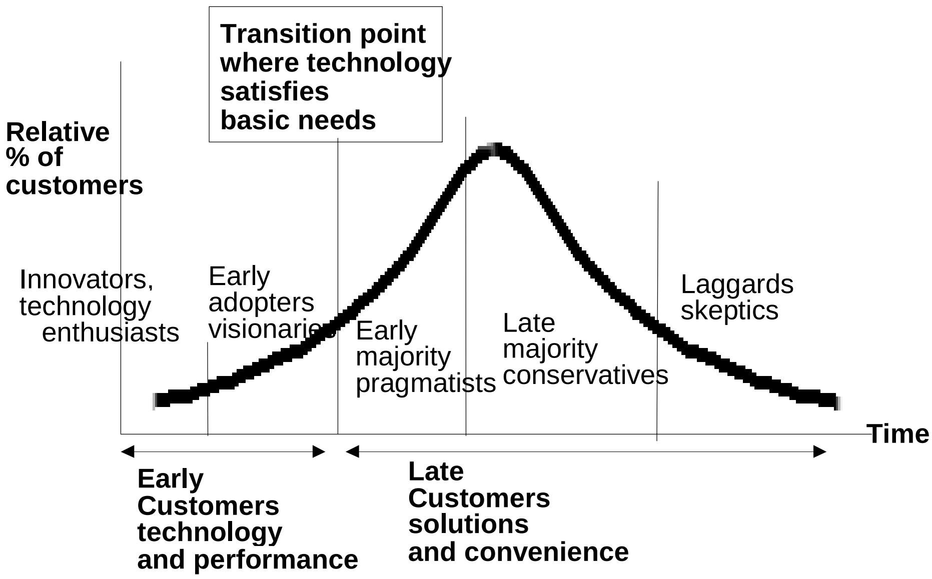

Figure 5.8. The change in customers as a technology matures and the change from technology-driven products to customer-driven, human centred products.

5.6.3 User segments and lifestyle segmentation

Winograd (1995) recognized three phases of computer product development: Technology-driven, productivity-driven and appeal-driven. Many current consumer products can be seen clearly appeal-driven rather than technology- or productivity-driven. For example, digital watches are not often sold based on the technology or functions of the product, but the product marketing tries to approach different consumer segments by emphasizing, for example, quality, trends, price, brand, lifestyle, emotions or personalization. Also mobile phone industry has entered appeal-driven product development, where the products are developed and marketed to consumer segments.

> *Segmentation is the process of partitioning markets into groups of potential customers with similar needs and/or characteristics who are likely to exhibit similar purchase behavior (Weinstein 1994, 2).*

Classification of users and potential customers can be based on buying behaviour and lifestyle. As in many technology areas, emotions and feelings are becoming arguments for selling and buying, and as drivers for developing better user interfaces (Nokia 2000b, Keinonen 2000) because the relationship between a user and a mobile phone often contains a lot of emotion and personality, maybe even more than the relationship of the user and a PC (Kopomaa 2000). Korhonen (2000) discusses *strategic design* where the design is explicitly started from a specific user group, or segment, and the product concept is developed to meet the needs of this group.

Mobile phone industry is developing products for different consumer segments. Marketing literature defines central variables for segmentation (Kotler 1997, Weinstein 1994):
- demographic: age, gender, occupation, income, education, family size
- geographic: region, city/metro size, density, climate
- psychographics: lifestyle, personality
- behaviour: benefits, user status, usage rate, occasions
- socio-economic: income levels, social class
- product usage: consumption levels, for example, heavy, medium, light, non-users
- benefits: what factors weigh in selecting a product

It is relatively easy to collect and use segmentation knowledge (demographic, geographic and behavioural) about customers using typical market research methods. In the real life, customers follow identifiable *lifestyles and trends* that change along the time.

In the competition, manufacturers and vendors have found it important to gain more specific information and understanding than segmentation knowledge on user populations and trends in order to enable *lifestyle segmentation*. In lifestyle segmentation, *detailed* or *deep* knowledge is needed about user's real-life usage patterns. Lifestyle segmentation, through listening, understanding and satisfying market needs, is the objective of Nokia's product development and market strategy (Steinbock 2001, 268). In the beginning of 1990s, Nokia developed the first lifestyle segmentation of its target customers. The four most important segments were "posers", "trendsetters", "social contact seekers", and "high-fliers".

People think, act, and are active makers of their physical and social reality. Interpretive researchers claim that relationships between people, organisations and technology are not fixed but constantly changing (Klein and Myers 1999). Hence, people and knowledge about people changes in the long run, in contrast to knowledge about material and data. For example, a mobile phone has potential to change a person's communication pattern, even lifestyle and communication culture of larger groups (Kasesniemi 2001, 77). Social learning theory (Bandura 1986) shows that people learn new attitudes and behaviours by observing others' actions and the consequences of their actions. Hence, the lifestyle segmentation based knowledge, especially in the fast developing area of mobile communication, expires fast and requires constant updating.

5.7 Users with special needs

“Users with special needs” is a large and loosely defined group. It includes, for example, elderly, visually impaired, deaf, deaf-blind and handicapped users. The group of mobile phone users with special needs is rapidly increasing for two reasons. Firstly, elderly people (often late adopter conservatives) are adopting mobile phones and the existing population of mobile phone users is aging. Secondly, due to increasing mobile phone penetration, also groups with special needs become mobile phone users. However, it is not reasonable to continue the categorization (for example: novice users, casual users, expert users, users with special needs), because this would lead to groups with common members (for example, disabled can be also expert users), breaking the principle of correct classification (Bunge 1967,75).

According to several (Nokia internal) studies, elderly and people with disabilities do not want to be stigmatised by using a special mobile phone but they want to use similar mobile phone as users with no disabilities. Looking at the majority of users with special needs, there are no physical needs to make dedicated mobile phones for users with disabilities. Many of the standard functions of Nokia products support special needs, such as shortcut keys, adjustable voice and ringing volumes, and voice commands. In addition, the basic product, a mobile phone, can remain the same, and with different add-on accessories it is possible to expand the capabilities of the mobile phone. These kinds of accessories are, for example, mobile inductive loopsets, which allow smooth interaction between a hearing aid and a mobile phone. However, some specific user groups will still have problems. For example, some users with severe vision problems are not able to see the battery indication in signal bar of the phone display. Matero (1999) studied elderly and disabled mobile phone users and reported many severe practical problems and risks related to mobile phone use. He shows cases where it is not enough to extend the mobile phone interface with accessibility devices, but in order to overcome the interaction problems more integrated solutions, such as larger keys or display with better contrast, would be needed.

5.8 Tasks

An information device is used to perform a set of tasks, typically related to personal activities such as communication. Several methods, for example TAKD (Task analysis for knowledge descriptions, Diaper 1989) and HTA (Hierarchical task analysis, Shepherd 1998), can be used to identify tasks and the ways the tasks are performed. A task analysis tries to find out the user's model of the task, goals and needs of the user, information needed for the task, steps that are needed to accomplish the task, and interdependencies between different steps (Nielsen 1993, 75-76). Concrete task examples are often used for task analysis. For example, TAKD was developed to analyze and structure real-life tasks: "TAKD was designed to identify the knowledge required to do a wide range of IT tasks … and to be able to represent such knowledge at higher levels of abstraction or generality (Diaper 2001)." In spite of the efficiency and versatility of the method, one of its inventors, Diaper suggests that people should stop using the method, and use other methods instead (Diaper 2001). He found out that there is a philosophical gulf between requirements specification and software design caused by two different perspectives: requirements are primarily anthropocentric whereas design is computer centric. The same problem has been identified also in other studies (Blum 1994, Hakiel 1997). In addition, traditional task analysis methods that are intended to model larger system activities, tasks and steps, are not able to capture many of the smaller “micro tasks” of mobile phone interaction.

The basic steps to perform a task can be easily identified, but a deeper analysis soon becomes difficult, especially for mobile communication tasks. For example, the phone call was described in Section 5.5. User's tasks with mobile phones are very different from tasks that are made with desktop computers, VCRs, or information appliances, such as PDA devices. Mobile phone tasks are mostly related to person-to-person communication in a specific context. A major difference between computers and mobile phones, due to personality in communication is the effect of affect. Affect refers to "a class of mental phenomena uniquely characterised by a consciously experienced subjective feeling state, commonly accompanying emotions and moods (Gardner 1987)." Another major difference is the mobile use situations of a phone, the mobile context.

5.9 Mobile context

Context: the conditions and circumstances that are relevant to an event, fact, etc. (Collins English Dictionary1999)

When the tasks and product use of a mobile user are analysed, it is important to understand the *mobile* context of use. Mobile context has the characteristics that it is not possible to foresee where, when or by whom will the product be used. Hence, in product development the context can be handled by assuming that the product will be used by anyone, anywhere, and anytime.

The following examples describe some contexts of use for a mobile phone user:
- the user made a phone call to his wife during office work, vs. the user used his mobile phone to call for help in an emergency situation.
- the user made a phone call when walking in bright day light, vs. the user made a phone call when sitting in a taxi late at dark night.
- the user was freezing and wrote a text message when waiting for a bus at a very cold day, vs. the user wrote a text message when travelling comfortably in a train.
- the user was at home and heard the phone ringing, vs. the user was in a department store and didn't hear the phone ringing due to noisy environment.

These examples demonstrate that the context of use for mobile phone is different from simpler and often static context of use that is often considered in the design of information products and software systems. The context of use for mobile phone is built, for example, from mental orientation in a specific situation, physical environment, day of time, and even temperature.

All possible contexts of use for a mobile phone and user cannot be fully described. However, several methods are used for analysing and describing potential contexts of use. Keinonen (2000) reports about using *cartoon scenarios* (Figure 5.9) for communicating context of use for designers. The power of cartoon scenarios is in the capability to:
- communicate non-verbal information about the context of use,
- combine views about places, persons, and add explaining information,
- describe the role of the product.

Figure 5.9. Cartoon scenario used in product design (Keinonen 2000).

5.10 Summary

Mobile phone usability is an intricate and complex entity that consists of expectations and real experience, the phone design as an entity, including the external interface, and from using the services that are available through a cellular network. Mobile phone usability is determined by the product, task, user and context of use. A designer can have an effect only to the product design, but other areas, such as the network capabilities, are not controllable by a product designer.

When a mobile phone is designed and usability engineered, it should be made keeping the whole user performance in mind. It also should cover the very different user groups. In product development the nature of *overall design* should be understood and applied in the early design phases.

Mobile phone usability is an entity that is defined by, for example (numbers refer to layers in Figure 5.5):

- the *availability of network services* (cellular network, content services) (1).

- the *ease* of services (2).
- the *utility* of service (2)
- the *device as a whole*, including default configuration (3)
- the *readiness* to be used, including first-use situation (3)
- the *informati*on of device state displayed (3)
- the *usefulness and availability of user support material*. (4)
- accessories (4).
- the *interoperability* of devices and services. (1-4)

I have now defined the user interface, the external interface and the service interface as the three conceptual interfaces that a user needs while interacting with a mobile phone. For each interface I have defined the specific elements that are part of the interface. Having the three interfaces as framework I have discussed several aspects from where the product usability is formed. It is now obvious that in the development of information products, the product usability does not come by accident. In the next chapter I shall discuss, what can be the consequences of insufficient usability engineering.

6 The Need for Usability Engineering in Product Development

GPRS phones of some manufacturers are already on the market place. The GPRS features of those phones are hopelessly, even ridiculously difficult to use (Talouselämä 2001).

Numerous reports and articles continually report problems with the use of some smart product. The reported problems may be caused, for example, by the lack of product quality, difficulties when taking the product into use, or failures when some specific task fails. Even if the manufacturer has tried to develop an easy-to-use quality product and applied usability engineering or human-centred design in the product development these problems may occur. Why?

I can access my bank account through an ATM, a call centre or a web browser. Soon I will probably be able to program by video recorder through my mobile phone or with a wireless keyboard and the TV screen. In each case the same data and functionality is accessed through a range of very different devices. The goals may remain the same but the actions and effects are very different (Monk 2000).

The future looks challenging for product developers due to positive and negative experiences in the use of current products and expectations towards future products. If you are not able to program your video recorder with dedicated controls, can it be easier with a mobile phone?

With concept *insufficient usability engineering* I try to describe the situation where the output of product development is a product with quality of use failures. Following ISO 13407 definition, the product cannot be used by the user to achieve specified goals in a specified context of use. The concept is problematic because there is no such usability engineering method or combination of methods that reliably produces a product without any usability problems, i.e. the level of *sufficient* usability engineering is difficult to define. For example, a study was done where seven professional usability teams independently studied the same web site and reported the found problems. Each team reported different usability problems with only minimal correlation between findings (Molich et al. 1999). Hence, more useful concept definition, not aiming to perfection, is:

Insufficient usability engineering leads to system with unacceptable or considerable number of quality of use failures.

Sufficient usability engineering leads to system with no or minimal number of quality of use failures.

Insufficient usability engineering is a consequence of incomplete organisational capability to develop user-friendly products. In this chapter I shall study, what are the effects of insufficient usability engineering, especially from consistency perspective, in the product's structure, development organisation and user requirements.

6.1 Some lessons

The following examples describe lessons from Nokia Communicator development, and one view to the challenge of dealing with new technology and complex user requirements. I will not try to explain reasons for the described problems, but the lessons are demonstrating possible effects of insufficient usability engineering.

The observations from communicator design have shown some repeating organisational reasons why the usability of product is not as good as it could be, and the usability engineering has been insufficient:

- the product development team has no understanding how the usability of the whole user experience is verified. There is no coordination of usability work, i.e. usability planning.
- there is no clear responsibility for product usability.
- usability engineers are not part of development team, but rather in-house consultants.

6.1.1 Consistency in browser design

A case study of smart phone consistency challenges is given in Article I. This article focuses on the design issues faced when an application (WWW browser) is downscaled from desktop computer to smart phone (Nokia 9000 Communicator) user interface and the design team is treading between *internal and external consistency*. The product user interface has a global meaning for function "Back" across the applications and, according to product UI style, it must be applied also in the browser. During the development time of the browser UI, major desktop WWW browsers used word "Back" for backstepping in the navigation history. The article shows that there are often very practical reasons why the both consistencies can't be achieved. External consistency, i.e. providing functionality and behaviour that was familiar from PC-based browsers, would have been desirable. This was difficult, though, because of the physical interface constraints of the device. Furthermore, the fact that the operating system of Nokia 9000 Communicator (Geos) is not in common use on desktop computers makes it difficult to achieve external design consistency.

When smart information products are designed, the integrity of the device is important and external consistency is easily sacrificed for the sake of internal consistency. Such "local optimization" of the interface often occurs when an application designer has limited knowledge of what the end-user is familiar with (Grudin 1989) or, for example, when the organisation *decides* to promote novel design ideas more than existing designs.

6.1.2 Usability failures and media

Lack of usability as well as technical failures are often recognized early by the media, for example, in product review in a magazine or newspaper, having effect on the customers buying decision and attitudes. The second generation Nokia communicator, 9110 followed the law of increasing product variety by introducing several new functions and more complex software than the previous Nokia 9000. This made the product more difficult to use, especially by leading to problematic start-up situation with communication functions (Example 2).

The next generation communicator, Nokia 9210, introduced new operating system (Symbian) and considerably increased product variety from Nokia 9110. Although several user-interaction problems were solved along the new product some problems were still reported, especially about the start-up situation with services. In this example the problem is not related to product’s user-interface design as much as it is related to the product’s service interface:

> **Example 3:** *HOST OF HASSLES. I want access to the e-mail I already have. I read on page 187 of the manual, I must … determine whether my mailbox service provider recommends POP3 or IMAP4 ... Help! What I want for my wireless e-mail connection is to click 5 or 6 OKs and let the computer do the work. (Businessweek, July 30,2001)*

The increasing complexity in product structure potentially leads to user problems especially in the beginning of product use. This is a problem for both the user and the manufacturer. Looking from these two perspectives the problem can be minimised by:

- keeping the user interface and actions required by the user consistent between products (user perspective).
- being constantly aware of the increasing product complexity when designing and building user support material for the first use situation (manufacturer perspective).

The balance of product factors is important. Although a new and innovative product may have problems directly or indirectly related to usability (example 3), it can still be a market success simply because of good product concept, as seen in example 4:

> *The rectangular-shaped Communicator, Nokia's all-in-one personal data organiser and mobile phone from Finland, has surpassed Palm as the most popular handheld computer in Europe. And, it's set to hit U.S. store shelves by early summer. (Reuters 2002).*

6.1.3 Introducing new technology in a product

GPRS is an example of new technology brought to consumer products. The claimed main benefit of GPRS is the capability to be wirelessly on-line all the time and high data transfer rate. To design functions based on GPRS requires large amount of technical knowledge, but also understanding complex user requirements related to the situation of being on-line in a mobile context with high data bandwidth. To be on-line is not anything new for those users who have an Internet connection, for example in the office. When the same concept, being on-line, is transferred to mobile context the first design problem is related to cost of the connection.

In tthe current European service model, whenever the user is on-line without wires, s/he is paying money for the connection. With some operators, on the contrary, the billing is based on monthly charge. Due to different operator-dependent service models, the user is required to be intelligent enough to understand when s/he is paying for the service and when not.

Helsingin Sanomat ("*Salalaskutus uhkaa gprs-verkossa - Puhelin voi siirtyä käyttämään minuuttitaksaa lähes huomaamatta* ", January 11, 2001) reported about new mobile phone product with GPRS function. The product design led to the situation where mobile phone user could be billed much more than was expected without sufficient indication in the phone user interface. Phone manufacturer claimed that the problem is caused by the operator billing system. Cellular operator commented that they have nothing to do with the phone functionality or product documentation.

The examples above give us understanding about some challenges of introducing new products, technologies and services to consumer markets. For example, the manufacturer can not know beforehand how the operators will bill for a service, or how the operator-specific set-up process will be supported by the operator. For some design problems this kind of shared information would be useful and helpful to all parties (user, manufacturer, operator). The acceptance of new technologies is dependent not only on the product design itself, but also on the provided service.

The examples present us an emerging area of usability engineering. It is no longer enough to usability engineer only the product, but the development of mobile communication devices requires building the whole user experience, including product's external interface and service interface. The most critical problems of using information centric products are met during the first minutes after taking the product into use.

6.2 Consistency

Consistency is important in such information products that should be instantly ready to be used and that should provide simple use and learning in any context. It is often mentioned as one of the most important and desirable design goals because it is assumed to improve the user's possibility for transfer of skills from one system to another (for example Shneiderman 1998, Nielsen 1993, Nielsen 2001). Still, user interfaces of today are often inconsistent (Nielsen 2001). The concept of consistency is difficult - even a team of 15 experts was not able to produce a definition for consistency (Nielsen 1989). Grudin (1989) criticizes that design emphasis on user interface consistency directs the designers' attention away from users and the context of use. Consistency should be maintained, not only inside one product, but also between current and earlier systems (Nielsen 1993, 90).

> *User interface consistency is used in three interrelated senses: the internal consistency of a design with itself; the external consistency of a design with other interface designs familiar to a user; and an external analogic or metaphoric correspondence of a design to features in the world beyond the computer domain (Grudin 1989).*

Smart phones have high number of functions and applications embedded in a compact form. Hence Smart phones are sometimes also called feature phones. The high number of embedded functions leads to the challenge of designing an internally consistent user interface. *Internal consistency of an interface design* is a goal that can be achieved by the product designers. In its simplest, internal consistency means similar menus, terms and action patterns. A project team can improve internal consistency, for example, with style guides and design coordination tasks. Internal consistency has positive effect on learning, ease of use and perceived quality.

External consistency of interface functions with functions of other interfaces familiar to the users is a far more difficult goal to achieve. External consistency can be achieved together with internal consistency if the interfaces are similar enough, but external consistency is often achieved at the expense of internal consistency. External consistency helps the user especially with the initial learning of the system by enabling the user to apply skills learned with earlier systems. When a product design adopts functions from other interfaces, the design team needs to decide which is more important, internal or external consistency (for example, Article I). This design issue is becoming an important issue in those design areas where new functionality is created by combining existing technologies. In the design of smart products the difficulty in achieving external consistency is also in the fact that the physical interfaces in the products are very different, for example a digital camera vs. a mobile phone, or a FM radio vs. mobile phone.

Nielsen (2001) lists the different levels of consistency expanding especially on the characterisrics of external consistency:

- the individual application
- across a product family
- for all products released by a vendor
- for all products running on a specific computer
- for the national computer industry
- internationally (for everybody, everywhere).

Correspondence of interface functions to familiar functions of the world beyond computing need to be analysed and designed when such metaphors, analogies, attributes or relations are applied that originate from non-computer interfaces. For example, when metaphor "scissors" is used in a drawing application.

The user is not able to understand different kinds of consistencies as well as a designer can. According to Grudin (1989), where the designers see two applications with internally consistent but externally inconsistent interfaces, a user may see one internally inconsistent system.

6.2.1 Converging technologies

> *Question: What do you get when you cross a computer with a camera?*
>
> *Answer: Computer (Cooper 1999).*

When new technologies converge in smart devices or new functions are added, there is always the question how internal or external product consistency is affected by the changes, or which consistency should be maintained. For example, if a digital camera is embedded in a smart phone, how much of the design should be consistent with digital cameras and how much with mobile phones?

Sinkkonen (2001) notes that systematic development of learnability is essential for developing mobile devices that are easy to take into use. One problem solving strategy in learning to use a new system is analogical reasoning. Every user applied this strategy in the beginning of learning process. Analogical reasoning in product use means that the user uses his knowledge and mental models from another product as a source and by imitating that, solve the target problem (Thagard 1996). Analogical reasoning leads to results if there is enough consistency between earlier experiences and current learning task.

Most of the functions that are designed to a smart phone exist already in some other system. For example, email, browser, text editor and time management applications are familiar from desktop computers, and imaging capabilities from digital cameras. Due to the different user interaction elements and mobile phone capabilities the functions and applications very often need to be redesigned and simplified for a mobile phone user interface.

Although people are highly adaptive to new platforms, their previous experiences are, nevertheless, critical in shaping their expectations (Norman 1988). For example, a web browser on a new platform should behave in a reasonably similar manner as browsers in other environments, to achieve *external consistency*. On the other hand, few devices are built for a single application only; the browser should also work in harmony with the other applications to guarantee *internal consistency*. The designer is often faced with a difficult design issue: how to weigh the different factors, or design constraints, that point in different directions.

It is precisely for this reason why consistency is a somewhat controversial design principle. While it is generally agreed that consistency is a worthwhile goal to strive for (Preece et al. 1994, Shneiderman 1998), many authors (Grudin 1989, Grudin 1992, and Norman 1988) warn about the pitfalls of taking only one aspect of consistency as the guiding rule in design. The overall goal, of course, should be to produce a usable product.

Monk (2000) uses terms *action-effect consistency*, i.e. consistency in the effects of actions, and *task-action consistency*, i.e. consistency in the way actions relate to task goals. Action-effect consistency is the principle that if the user takes some action it should have the same effect whatever the context, i.e. the interface should be mode free. For example, double clicking on a word should have the same effect in all parts of the user interface. In small devices, such as mobile phones, there are inevitably several modes, because there can be a large number of commands but only few buttons to access those. The action of pressing a button will have different meanings depending on the mode the phone is in. This leads to *action-effect inconsistency* (modedness), which may be acceptable if the user is aware what mode he is in. Task-action consistency is intended to result in transfer of training when learning the set of actions needed to achieve similar goals. Similar goals should require similar sets of actions to achieve them.

Function richness and redesign of applications are challenges for the design of new smart phones, for example, because:

- overall product complexity increases (new functions, new technology)
- product size decreases (smaller and/or less interaction elements)
- new technologies are embedded
- new technologies are merged together (new functions are created based on the opportunity that the merging of technologies provide)
- functionality is downscaled from desktop computers and other systems to mobile phones (familiar tasks are redefined and renamed)

In order to provide an externally and internally consistent user interface, the designer needs to have sufficient knowledge about:

- what kind of consistency is preferred?
- general and product specific design guidelines
- design of other parts of the user interface
- dominant designs of similar functions and possible sources of transfer-learning
- the characteristic and opportunities of the new available technology, or about the merged new technologies.

The lessons from the development of three generations of communicators have shown that:

- basic and logical function consistency is required between systems. For example, certain functions of text messaging application needs to be available device independently in order to meet the user expectations and requirements.
- when the product is an integrated entity, the internal consistency is more important than external consistency. For example, users may not expect similar menu structures in a hand-held device and a desktop computer for the same application domain.
- even if the internal consistency is an explicit design goal, it is difficult to fully implement due to product's structure
- product should be inter-consistent enough to provide easy switch from one product to another.

6.2.2 Development organisation

Nokia launches several new phone models each year. Increasing number of the phone users are replacement users, i.e. users already have a mobile phone and they want to replace it with new one. Currently more than 70% of the customers are replacement users (Gartner 2001). The importance of inter-consistency is high, because users do not like learning new ways of using the new phone. Also the need for product consistency is high because users expect to be able to use new functions in the same as they use the familiar ones, and to quickly learn the new functions.

If the product development organisation succeeds in the product design, it creates a product that has external consistency with other products and internal consistency between the new functions and old functions. If usability engineering is insufficient there is a high risk that *both* internal and external product consistencies are poor.

A special challenge for the development organisation is to develop internally and externally consistent products in the multi-site development environment. External consistency can be difficult to maintain, even between company products, if the projects or whole teams are progressing in different continents. To develop a product with internal consistency is challenge for multi-site teams, where the individual designers of one team are located in different places. The effect of insufficient usability engineering can be seen also on practical level of project performance. The following table (Table 6.1) lists some observations made in the development of smart phones and communicators.

	Sufficient usability engineering	**Insufficient usability engineering**
Design creation	Design and improve by iteration	Late rework needed
Schedule	No negative effect	Timetable pressures due to rework.
Designer support	Faster problem solving, more confident about decisions in the form of decreased design uncertainty. Awareness of the design goal.	Difficult to verify design decisions. No awareness when the design is "good enough". Design uncertainty.
Management support	Awareness of expected user problems.	No realistic understanding about expected user problems.
Usability engineering costs	Can be predicted	Not fully predictable due to costly changes and rework in the late development phases.

Table 6.1. Some comparisons on the effects of sufficient and insufficient usability engineering.

6.3 Possible levels of usability engineering

It is not self-evident that product development does or need usability engineering. The interest or capability is dependent on the product, project, organisation, resources, and product development processes. The three examples above (in Section 6.1) show, however, that there is a high risk of developing products with considerable quality of use problems when the design has internal or external design uncertainties

In order to analyze, create synthesis or to compare (Bunge 1967, 74) why one project created a usable system while the other did not succeed in that, it may be useful to establish a conceptual tool for measurement. For a smart phone project we can define the possible levels of usability engineering based on the coverage, duration, or visibility. The following lists give 0-4 ordinal scale (Järvinen 1999, 101) for each level.

Usability engineering coverage defines which areas of product development are covered with usability engineering activities.

0. No usability engineering
1. Usability engineering on limited issues of some design area, for example with one SW function.
2. Usability engineering covering one entire design area, for example software
3. Usability engineering covering many, but not all, design areas.
4. Usability engineering on all product design areas (full implementation of human-centred design).

Usability engineering *duration* defines the temporal execution of usability engineering activities.

0. No usability engineering
1. Usability engineering at the late development phase
2. Usability engineering at the early development phase
3. Usability engineering during the whole project lifecycle
4. Usability engineering covering several project lifecycles (several products)

Usability engineering *visibility* defines the extent to which a usability engineer is capable to participate the product development project.

0. No participation in the project
1. Subcontracted/paid usability test ("black box", company external usability engineer with no access to development team)
2. External usability consulting (company external usability engineer with access to development team)
3. Usability engineer in the company (company internal, project external)
4. Usability engineer in the project (project internal usability engineer)

The levels can be organised in triples [Coverage, Duration, Visibility] to have a sufficient view to the way how usability engineering is integrated to development. Based on the classification, it is now possible to define the optimal usability engineering case (4-4-4) and the worst scenario (0-0-0). Problems related to insufficient usability engineering can be analyzed based on this classification. For example:

- visibility <3 (for example [2,3,2]): This is a typical consultant case. The project uses a usability consultant to verify product usability. A consultant can typically provide training, product analysis and testing of product functions, but the impact of proposed changes may be low. A consultant is effective and useful especially for training and temporary problem solving for a defined period of time.
- visibility = 3 (for example [1,2,3]). If the usability engineer works for many projects there are practical challenges in performing long-term and continuing activities with one specific project. When visibility is 3 the usability engineer can help in solving a specific design issue, but is not capable of serving many design areas or the whole project. However, knowing many projects is an advantage in maintaining inter-consistency between products due to wider hands-on perspective.

6.4 *Coping with product structure*

Due to complex product structure and dependencies between concurrent design areas, insufficient usability engineering is seen as low capability to
- support the product design integration,
- identify and to make good use of new design opportunities and innovations,
- make new functionality and functions usable
- evaluate the design proposals of *innovative* functions.

In the physical product design new functions often require re-organizing the mechanical and hardware structure. It is not possible only to plug in new components. In the compact design of information appliances structuring can have an effect on the user interaction elements. For example, there is less space for keyboard, position of microphone and loudspeaker must be rethought, new functions require new interaction tools or change the behaviour of familiar ones.

The increasing complexity of product structure and integration of new technologies does not only mean that there are more functions. It means also that the behaviour of old functions may change. For example, traditional Phonebook is not anymore a phonebook, but it is also tool for managing mailing lists and keeping track of personal communication ("log"). This leads to problems that are related to learnability (consistency, transfer learning, analogical reasoning), user errors and predictability.

6.4.1 Operating system

There is not yet an optimal operating system for a smart phone. When the operating system for the first true smart phone, Nokia 9000, was considered it soon became obvious that the (Nokia proprietary) operating system that was used in earlier Nokia phones would not work in more demanding environment. Hence, Geos was selected for the communicator operating system in 1997. Along the further development and increasing product variety it turned out that Geos can not efficiently support the tasks and more complex communication functionality, and it was replaced with Symbian operating system for Nokia 9210 in 2001.

Other phone manufacturers are trying to cope with the same problem. Attempts to turn Palm or Microsoft software platforms to mobile phone operating systems have not been successful this far for various reasons. For example, the operating system may be ideal for PDA functionality, but it doesn't have enough capability to perform required signal processing computation of a cellular phone, or the operating system requires more memory than can be built in a compact hand-held device.

Operating systems are changing and evolving fast, and there is no reason to believe that the situation would stabilize. Today, almost each product category and generation has different operating system from the predecessors or competitors, and the situation seems not to stabilize. In smart phone design we will be over and over again in the situation where a new operating system is introduced along a new product. The challenge for usability engineers, who might not have wide knowledge about software, is to understand the potentials, limitations and future evolution of operating systems.

The continuously changing and inconsistent operating system (between products) is a huge challenge for the users and the designers. Even though the operating system should be invisible for the user, it often sets limitations and characteristics for the device performance and use.

6.5 Product requirements

Lederer and Prasad (1992) studied large software projects that significantly overran their cost estimates. The study showed that the four reasons with highest responsibility for inaccurate cost estimates were related to usability engineering: frequent requests for changes by users, overlooked tasks, *users' lack of understanding of their own requirements*, and insufficient user-analyst communication and understanding. Nielsen (1994, 5) claims that proper usability engineering will prevent most such problems. Requirements are gleaned from a variety of sources, such as help-desk call –log analysis, market research, product reviews, and user groups with direct or indirect contact (Keil and Carmel 1995, Sawyer 2001). Product requirements for a mobile phone originate from three main sources: the company, user, and cellular operators.

6.5.1 Manufacturer requirements

A company developing smart phones needs an understanding about the future directions concerning new products, technologies, functions, and other dynamic factors defining the expected market success and demand. This understanding is based on different kinds of information sources, such as market research, product roadmaps and customer processes (Valjus 1994). On a product level the understanding is seen as technical product requirements that are needed in composing the product specification. "Usability" as a manufacturer requirement is often a general goal, easy to express but difficult to measure or assess.

6.5.2 End-user requirements

For mobile phones, or any other consumer products, there is no such a user population or segment that can fully represent all users. Hence, to elicit the end-user requirements for a new product requires use of several methods, such as user requirement analysis, understanding market behaviour and trends, having continuous contact to users, and intelligent guesses when innovative functions are designed.

End-users have expectations towards new product based on the previous or competing products. Hence the difference between old and new product concept, or the innovation type introduced in the new product, can be used as parameter when evaluating the potential user problems or the success in meeting end-user requirements.

End user requirements are turned to design documents having effect on both the product functions ("feature list") and user interface specifications. End-user requirements are often concrete functional, qualitative or quantitative entities that the design team can use as a goals in creating the product design. Usability as end-user requirement is typically related to a concrete functionality that can be measured and assessed during the product development.

6.5.3 Operator requirements

Most mobile phones globally are sold by cellular operators. The operators have fact-based knowledge about the use of different product functions. Operators also have understanding of what kind of products they want to sell in order to generate as much cellular communication traffic, “air time”, as possible.

Operators have their own product requirements. These requirements cover technical product quality, functions and also usability. The requirements are assessed against launched product candidates and sometimes negotiated with manufacturers during product development in the form of technical testing and usability testing. If the manufacturer product does not fill operator requirements, the operator may require product changes (costly at this point), or refuse to take the product in its portfolio.

6.5.4 Summary of requirements

Functions of a smart phone are mostly related to voice or data communication. The important factor for user and operator is that wireless communication via cellular networks always costs money. To understand user requirements, requires understanding the hidden or visible costs related to the use of a new function (GPRS example in Section 6.1). Users want to minimise costs, operators want to maximise the revenue, and manufacturers try to serve both.

Each product development project gets hence different kinds of requirements. Only some requirements are defined by the development team. The quality and level of requirements vary, and some requirements can be conflicting with each other. The interpretation and turning of product requirements to design are difficult tasks. They require capability to analyse, refine and consolidate the requirements. Many requirements can be analysed with usability engineering methods.

Requirements are normal outcomes of several user-centred design and usability testing activities. When requirements are properly documented, they accurately detail what the system must do. However, these *behavioral* requirements never cover all system requirements, such as data formats, external interfaces, business processes or performance targets (Cockburn 2001). Sufficient usability engineering can help the development team in identifying, analysing and managing product requirements and user requirements, and to solve challenges related to conflicting and vague requirements. In the worst case, insufficient usability engineering can lead to development where the changing requirements increase the development time due rework in product design.

6.6 Summary

When a new function is designed or an innovation is implemented, there is an apparent need for usability engineering. On the one hand, the designer should have knowledge about the user needs, and on the other hand the designer should understand operator needs with specific function or the particular business area.

> Insufficient usability engineering leads to incomplete understanding of product requirements, difficult-to-use functions, potentially (internally or externally) inconsistent product, and product that is difficult to take into use.

In a development project there are sometimes practical restrictions why the usability of a design cannot be verified as it should be. Typical reasons for these restrictions are optimistic project timetables, lack of resources, lack of sufficient user testing during product development, and incomplete knowledge about the use of the product. Innovative products are constantly in the situation where it is almost impossible to simulate the context where the product will be used.

In this chapter I have studied problems that may be caused by insufficient usability engineering in the development of a mobile phone and discussed some potential reasons for the problems. In order to solve the research problem of this thesis, we are first interested in to understand the reasons that led to insufficient usability engineering (this chapter), and secondly to form solutions through real product development trials that help us to avoid the same problems from occurring again (next chapters).

7 The Usability Plan as Part of the Concurrent Engineering Process

Due to the complexity of product structure, organisation, user requirements and high degree of interaction in the product use, it is not an easy task to develop a usable mobile phone, or guarantee that customers will accept or want the new product. Designing for the usability in complex consumer products such as smart phones, it is often impossible to cover the entire product design (the user interface, the external interface, the service interface) with available usability resources. In addition, due to the competition the project timetables are always aggressive.

Because of development complexity and tight timetables, a successful project and respectively successful usability engineering requires good management principles to be used: careful planning, prioritising the tasks, managing risks, communicating effectively and delicate synchronisation of concurrent activities. For product design this means that design groups (for example hardware, software, industrial design, and user interface teams) need to be well aware of the dynamic status and targets of the other design areas during the whole project lifecycle.

In this study two specific approaches for ensuring the usability of the developed product, i.e. to avoid insufficient usability engineering, were applied: Usability Planning and Usability Risk Management. In this chapter I shall describe and evaluate the success and usefulness of the two approaches.

7.1 Overview of planning

When a development project is long (years) and the system to be designed and implemented is complex, it may be difficult for a usability engineer, even if member in the development team, to systematically concentrate on and to keep continuous focus on the most problematic design issues, and to efficiently react to emerging design challenges in parallel design areas. In these kinds of projects in general, a good planning is needed for a successful project. Planning tasks are, for example:

- defining goals, milestones and tasks
- defining effort estimation and schedule
- planning of risk management

Lack of planning may lead to wrong timing of actions, lack of intermediate goals, blindness in seeing and capturing the main problems and bottlenecks, and finally unsuccessful project output, such as major delays or a system with quality failures. My informal discussions with usability engineers in several industries (for example, a mobile phone manufacturer, a bank, three SW development houses, a new-media company) indicate that even though many product development projects do usability work, and in many cases good usability is defined as an explicit goal, the projects still do not properly plan the usability efforts for project lifecycle or define what the required "good usability" means. This produces similar problems to those created by the lack of planning - reactive actions when problems emerge instead of proactive analysis and design, for example.

In order to successfully perform usability engineering for a complex product, we need to plan the usability work. This is needed to provide efficient, well-timed and focused usability engineering that supports the whole project execution, i.e. project management, system design and implementation, and is in synchrony with the project. The problem is, how to support a project in developing a successful and usable system that has potential for market success and customer satisfaction?

7.2 Some roles and perspectives

In typical concurrent product development, product designers do not have an extensive view of markets or hands-on access to customers and their needs. Each designer designs the product horizontally from an own perspective or discipline, with the target of providing a robust product component for product integration. The designer has a limited understanding about the expected product entity and limited view to activities on other product development areas and disciplines (Parker 2001). A usability engineer needs a good understanding about the product entity, knowledge about product usability problems, and solutions for eliminating the problems. A usability engineer needs methods to:

- build a holistic view of product design already in the early design phases
- verify that user requirements *will* be met
- verify that product's user interface, external interface and service interface are usable.

Designer, usability engineer and product manager have different perspectives and foci to the product development (Figure 7.1). The perspectives are exemplified (and simplified) with the following questions:

- project management perspective: How do I take this *project* to end?
- designer perspective: How do I solve this particular *design problem?*
- usability engineer perspective: Will the product work as the users expect it to do?

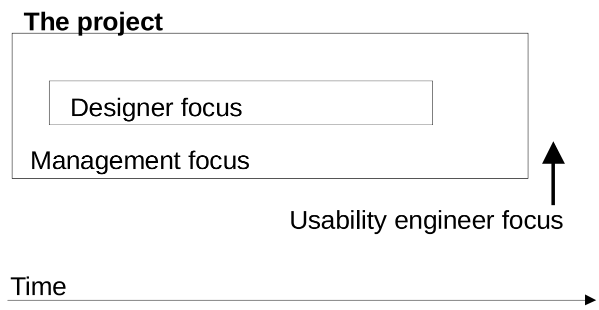

Figure 7.1. The working focus of designer, usability engineer and manager.

A designer has focus on the product development within project lifecycle with limited view to other parts of the development than the own discipline. A designer aims to provide agreed deliverables at different phases of the project. A usability engineer looks at and tries to involve in the various product development activities across disciplines. The focus is on meeting the user requirements in the final product. In a way, a usability engineer looks at the moment that comes *after* the product development. Project management has a mission to steer the project so that the product is ready within agreed timetable and with promised functionality.

7.3 Approach for usability planning

I used a systematic method to plan and perform usability engineering in a smart phone CE project. The usability planning method was repeatable. Other product development projects started to use the method in parallel with the case project after a successful start. The method was quickly improved to meet the needs of different kind of development projects. This approach provided many advantages. Firstly, a common and repeatable usability planning method of CE projects was established. Secondly, using the same method in parallel projects provided us a way to assess and improve the efficiency and usefulness of it. The objective was to identify project management methods that can be applied with human-centred design methods in order to manage and focus on usability engineering in a CE project. The exact research question is, is it possible to develop a method for planning usability work in CE project? The result is a method with concrete task descriptions for defining the project Usability Plan compatible with CE practices. This method improves the method presented by Mayhew (1999) by introducing risk management, timing and prioritisation factors. It also adds understanding about the early and late development phases in CE.

The findings show that a Usability Plan for a development project should be done together with project management in very early phase of product development in order to support and take advantage of CE. I also found that several project management tools can be applied in usability engineering.

7.4 Concurrent Engineering opportunities for usability engineering

CE provides several opportunities for usability engineering and planning of usability work. Well-defined *project milestones* provide the rhythm for usability engineering. In each milestone it is known:

- what design products are available or possible (prototypes, design details),
- what design products are not yet available,
- what can and should be done *before* the milestone in order to improve the product design
- what can't be changed *after* the milestone is past.

For cost-efficiency and design-efficiency it is important to understand what kind of work is needed in the different design phases. The following figure (Figure 7.2) shows how many design changes could be made in different phases of the case project. I created the data by interviewing 8 designers and 8 engineering managers of the specific 8 development areas (concept definition, industrial design, user interface design, mechanical design, software design, hardware design, localisation, user documentation) using a structured questionnaire. The participants were asked to verbally describe the current situation and to draw the estimation of doable changes in a figure that describes the milestones. The answers were later validated by studying the proposed and implemented changes, in comparison to the answers of the questionnaire and knowledge from previous projects. Any major contradictions could not be found. Y-axis describes the estimation for “how many changes can be done (or was possible to do) to current design in milestones”. A scale from 0 to10 was used. “0” expressed that no changes can be done (no design flexibility), and “10” expressed the situation where any change is possible (full design flexibility).

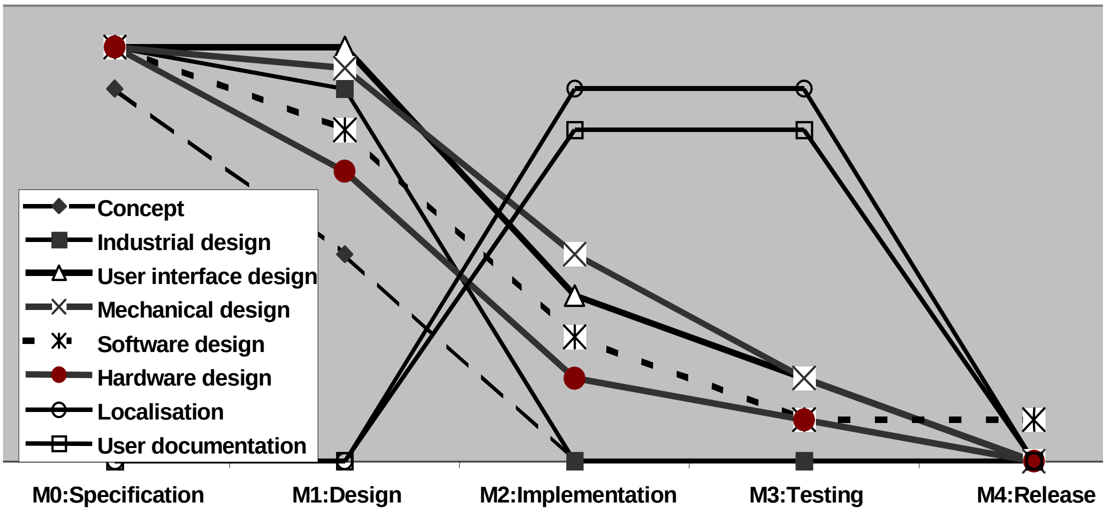

Figure 7.2. Product design changes in CE milestones.

This chart provides several important findings:

1. Most product design (industrial, user interface, mechanical, software and hardware design) changes are made between M0 and M2 with the principle that "harder" design areas are frozen first. The product concept is the first to be frozen.
2. Two patterns can be detected. First, main engineering areas (software, mechanics, hardware) freeze gradually from M0 to M2. Secondly, supporting engineering areas (localisation and user documentation) have main design phase between M2 and M3 with faster freezing speed.
3. Some changes are made even in late design phases (software, localisation, user documentation).
4. The turning point from early design (creating the design) to late design (design changes and improvements) can be estimated. The turning point can be seen at M2.

Milestones are places where the performance and status of the project are carefully reviewed in order to decide whether the project can continue to the next milestone. *Milestone reviews* provide natural occasion to communicate effectively the results of usability engineering this far to the project management and design teams. The qualitative and quantitative knowledge obtained from usability engineering activities improves the understanding of project quality and design maturity.

As in any large project, the estimation of timetables is difficult to do and often too optimistic (Keil et al. 2000, Premkumar and King 1994). In CE project the estimate for *next* milestone X is the most accurate information available, and estimates for other milestones are dependent on this milestone. It is reasonable to plan issues that must be done before the next milestone and to set exact timetables only for the nearest period instead of planning a weekly timetable for a longer period.

If the CE process is well established in the company, it means that there are viable *project management and product design processes*. These processes can be applied and adopted in usability engineering as well. For example, for project management there may exist a process or practices for managing development risks, or for SW design there may exist a process for managing change requests after design has been frozen.

7.5 Usability Plan

A Usability Plan for CE project defines the project actions that aim for achieving the required product usability. In this study I defined and tried out a method for planning usability actions in a smart phone project. The planning of usability actions for a CE project was based on agreeing with the project management about project and usability goals, setting usability engineering priorities, timing the actions according to project milestones and maintaining usability risk prediction data. Usability Plan aimed at supporting both project *management* and product *design* work. Principles of human-centred design and ideas from usability planning ideas (Mayhew 1999, Daly-Jones et al. 1999) were built-in in the plan.

Usability Plan is based on differentiating coordination tasks from performance tasks. *Performance tasks* form the actual usability engineering, such as usability tests and design iterations. Performance tasks need *coordination tasks* to guide the project usability engineering, such as definition of usability requirements, timetables and project reviews.

7.5.1 The birth of Usability Plan

In the beginning of the case project I needed to analyse how usability engineering could have been done better in the earlier observed projects (Figure 1.2) and what were the main problems for a usability engineer in these projects. This analysis led me to ask the following questions:

1. What can we learn from the earlier projects? What kind of usability problems these products have? What usability activities have been done and what is their effect in the product design?
2. What issues (organisational factors, product functions/interaction/capabilities/technologies) are most important for the success of the case project and product? Can we derive usability goals from these?
3. Is it possible to identify design uncertainties or risks in the product concept and new functions? What are those and how can usability engineering solve them?

When these questions (1-3) are answered it is possible to continue to the questions that support the actual usability planning. I studied each project milestone (M0-M5) with the knowledge of typical milestone content, and analysed from coordination and performance perspective:

4. What usability activities are *possible* before milestone X?
5. What usability activities *must* be done before milestone X?
6. What usability activities are *possible* and reasonable after milestone X?
7. What usability activities *must* be done after milestone X?

The understanding and insights that were gained through the questions were documented in the first version of a Usability Plan.

7.5.2 Usability assessment meeting

Each product has a reason, or reasons, why it is developed. The reason can be, for example, introduction of L-, E- or I- innovation, company strategy or roadmap, fulfilling market demand or competition.

Anderson et al. (2001) proposed that a high-level agreement is an essential step towards integrating usability into company’s process. This was implemented as Usability Assessment Meeting. Usability assessment is a meeting in the very early phase of the project where project management and usability engineers share the product motivation and identify the usability-related issues critical or essential for the success of the product. The meeting is a two-way starting point for product usability engineering and creation of the Usability Plan. On the one hand, product management can share in a structured way its needs and requirements about product quality for usability engineers, and on the other hand, usability engineers can gain sufficient project management support and commitment for usability engineering.

In the case project, Usability assessment meeting was held at the beginning (near M0) of the project. In this meeting the project management discussed the project vision and product concept issues and priorities together with the usability expert, and agreed on general framework and targets for usability engineering. This meeting provided a solid and documented background for building the Usability Plan.

Before M1 the project and product are still looking for the form they will take. A usability assessment made too much before M1 may be inaccurate or it can even give wrong and misleading information. Hence, information obtained from usability assessment requires updating and checking while the project proceeds.

7.5.3 Usability requirements, targets and priorities

Usability requirements and targets are standard usability engineering issues and widely discussed in usability handbooks and research publications. I extend this toolbox to contain also *prioritisation* of usability activities

Usability requirements, the primary objectives for usability engineering, define when the product usability is acceptable. Usability requirements originate, for example, from the project requirements and the Usability Requirement Analysis (Mayhew 1999). In a CE project, requirements can be defined, in addition to the product entity, for each design area and each milestone. Status against usability requirements is assessed in milestone reviews and at the end of project usability engineering.

While user requirements define what functionality must be available, *usability targets* (or goals) define quantitatively or qualitatively *measurable* goals that are possible to assess with usability engineering methods. These targets can be set for both subjective measurements (often questionnaires), qualitative results, and quantitative measures, such as counting text input time and errors. In the case project, target setting was difficult especially for issues without comparative or benchmark information, such as functionality related to new technologies (for example, efficiency in Bluetooth interaction between two phones)

CE projects often deal with complex products. This implies that sometimes the product is too complex to be usability engineered for every detail (or even for every application if there are many). I used a method for *prioritising* the product design areas for usability engineering. Prioritisation was agreed by project management and usability engineer. This made it possible to efficiently focus on the issues that are critical for the success of the product and customer satisfaction, instead of working in ad-hoc mode. It also enabled me to work in depth-first mode with important design issues. Prioritisation had two levels. Priority 1 issues were engineered at all phases, and priority 2 issues at selected phases of the project.

7.5.4 Planning matrix

In the case project I used a milestone-based planning matrix for high-level job division and task coordination. Usability planning matrix was created based on priorisation and the coordination and requirement information. Also the knowledge about design phases in different design areas (Figure 7.2) could be applied in this matrix.

Table 7.1 shows an example of the planning matrix. First, it defines what coordination actions are performed in project phases, for example Usability reviews. Then the Priority 1 issues are listed as usability engineering actions in all phases, and the Priority 2 issues are listed in phases where they are optimal for usability engineering. This matrix is an effective but simple tool for planning the overall usability work amount, and even for timing of the single tests in the different project phases and for maintaining an overall picture of project usability engineering.

Action	**M1**	**M2**	**M3**	**M4**	**M5**
Coordination: Usability review	X	X	X	X	
Priority 1: Phonebook	X	X	X	X	X
Priority 1: Navigation device	X	X	X	X	X
Priority 2: Text messaging		X		X	
Priority 2: Out-of-box readiness	X	X		X	

Table 7.1. Planning matrix for usability actions

7.5.5 Human-centred design in CE phases

ISO 13407 describes a human-centred design (HCD) process and the main activities in it. According to Article II the following HCD phases fit naturally to a CE project:

- early design: Collect information, such as field feedback, about earlier products. Use existing knowledge to develop design proposals. Make the design solution concrete using simulations.
- early and late design: Present the design solution to users and allow them to perform tasks
- (early and) late design: Alter the design in response to the user feedback. Iterate this process until objectives are met
- early and late design: Provide design feedback

The very early and very late development activities, such as user needs studies and field feedback analysis, are not typically in the scope of a CE project. Nevertheless, the organisation should be able to perform these activities in other functions than a CE project.

Kiljander (1997) provides several examples of user interface design evaluation in early and late design. However, there is no exact definition of the time when a design process changes from early to late design in CE process. In most CE product design areas and projects observed in this study the shift from early design to late design happens near M2 (Figure 7.2).

By definition, HCD requires design iteration. In a CE project iteration is performed during early design continuously across milestones. Practically, re-design (changing existing design) and iteration is not cost-efficient or often even possible during the late phases. In the case project, a standard iteration sequence was created. Early design always started with heuristic evaluation based on the design documents, continued to low-fidelity usability testing for formative feedback, and concluded with high-fidelity or product prototype testing for summative feedback in the late phases.

Early design aims to *create* a design. The purpose of usability engineering in this phase is to *support the design work* and evaluate design proposals. Early design is time-critical, because it provides the first potential bottleneck for the start of the implementation, and because the design tasks in different areas may be inter-dependent. The nature of the early design phase needs to be taken into account in the Usability Plan. Early design evaluation is tightly integrated into daily design work and should thus be resourced by the project staff, instead of, for example, external consultants. Usability engineering in the late design phase aims to validate and diagnose an existing design (coordination purpose), provides information to make the design better and analyses problems (performance purpose). Late design is not time-critical, because the project can progress with the existing design decisions and documents. The late design evaluation has more potential to be done by external resources, such as subcontracted usability tests.

7.5.6 Coordination of usability engineering

Coordination tasks define the overall project management and engineering framework, such as goals and milestones. Typical coordination tasks are planning, evaluation and follow-up. Coordination actions provide the information that project management needs for following the usability status of the product. Hence the coordination of usability engineering is a subtask of project coordination.

Usability engineering needs coordination for knowing the limitations and opportunities of the work. Coordination actions are often similar to those in other product design areas. Usability coordination issues are, for example:

- start and end of usability engineering
- input and output of usability engineering (what is needed in order to start it, what is its expected outcome?)
- required documentation (for example, review documents, test reports)
- responsibilities
- review, evaluation and management practices
- what resources (technological, knowledge, human etc.) are available and allocated?
- what is the budget?
- final and intermediate targets.

7.6 Usability Risk

The effects of insufficient usability engineering were discussed in Chapter 6 with the conclusion that insufficient usability engineering leads to incomplete understanding of product requirements, difficult-to-use functions, potentially inconsistent product, and product that is difficult to take into use. Norman (1998) discusses the related failures of products in "The Invisible Computer - Why Good Products Can Fail" with the explanation that many products fail because of technology oriented development instead of user-centred design. Markeset and Kumar (2001) claim that many product failures can be traced back to design engineers' inability to foresee problems that occur later in product life.

To some extent a product failure can be caused by such poor design, which could have been detected and corrected early in the product development. This potential reason for product failure is called *Usability Risk*. Usability risk is not related to technical product quality which can be detected in technical product testing, but it is an issue that emerges in product use, and leads the customer to abandon the product or brand. When a usability risk comes true it may lead to the failure of the product or failure in using a service related to the product.

Platt (1999) introduced the term Usability risk in the context of e-commerce and WWW services. Her practical interpretation of the risk is that the users will select the easiest and most convenient services: "…the online audience…do require a much easier and more inviting interface…there are plenty of people ready and willing to compromise their personal information and anonymity for convenience and discounts… We all know and care a lot about managing the technology risk, the market risk, and the money risk. We ignore at our own peril the usability risk". Snyder (1997) discusses look and feel. She says that SW products seldom fail due to poor "look", for example bad icons, but the "feel" is the area where the risk resides. According to Snyder, with judicious use of prototyping and usability testing it is possible to discover are we building the right thing.

The concept *front loading* (Blackburn et al. 1996) is related to prevention of usability risks:

> *Front loading is the early implementation in upstream design activities of downstream functions...It provides an early warning about issues that, not considered, could lead to costly redesign and rework later. In hardware, for example, front loading of manufacturability problems can keep them from becoming major show stoppers when the design is transferred to production. Front loading of information about the direction of developing technology can aid the creation of more robust, reusable designs.*

Because a usability risk is related to the user acceptance, or meeting user expectations in a specific situation or context, it may be useful to consider issues not typically covered by basic usability engineering methods:

- the user may experience a breakdown in the use of a new product. The breakdown can happen because the difference to an earlier product is too small (user prefers the earlier product and does not want to change) or the difference is too big (user is not willing to learn too much new functionality) (Vessey and Conger 1994).
- the user is not able or willing to see or identify the idea or concept behind the new product (computer self-efficacy[12], Marakas, Yi and Johnson 1998).
- the user's environment (culture, social or physical situation, context) has negative or positive influence on the acceptance of a new product or function (Bandura 1986).

7.6.1 Usability risk management

Project risk management is needed for verifying that a project reaches its targets. Usability risk management helps the project to reach its targets in terms of user acceptance by eliminating the problems critical for achieving them.

A usability engineer in the project has often a good overall picture of the product entity and functionality, especially in the early phases of the design. By usability engineering in different design areas the usability engineer finds out issues which may cause the product to fail in terms of user acceptance. This knowledge can be documented in the form of *usability risk prediction data*. Cockburn (2001) and Snyder (1997) present the basic risk reduction and management strategy (Table 7.2):

Cockburn, risk reduction	**Snyder, risk management**
Look carefully all around the project	Identify the areas of biggest risk.
Detect the risks.	Gather data
List the risks in order.	Revise the plan and product design as needed.
Work on them in order of danger.	Determine if you are on course. "lather, rinse, repeat"

Table 7.2. Risk reduction and management.

[12] Defined as "an individual judgement of one's capability to use a computer" (Compeau and Higgins 1995, p.192).

In this study the usability risk prediction data was maintained through the project. Risks were identified mostly in usability tests and by constant analysis of developed product against expected use situations and emerging user needs. This data described the severity and effect of the risk for each observed risk, and a plan for resolving the problem (Table 7.3). During the project this data was used to keep the usability engineering focus on the areas that were seen as usability risks. Thus, it was a tool for the usability engineer and also an information source for the project management. In several occasions this data was used for redirecting design efforts and resources. For example, when risks with an input device were identified, a cross-disciplinary team, "task force", of engineers was established to remove the risk.

Risk	Description	Effects	Severity	Probability	Cause
1	Input causes too many errors	High error rate	High	50%	Changes in text input algorithm

Table 7.3. Example about usability risk prediction data

7.7 Evaluation of Usability Planning and Usability Risk management

The quality of the presented method for planning usability engineering in CE project can be evaluated conceptually and empirically, the basic criteria being that a method is good if it solves a problem (March and Smith 1995). In this case the problem is: how can usability engineering be planned in a CE project?

Planning benefits are difficult to assess using objective measures (Premkumar and King 1994). Perceptual measures such as using the fulfilment of planning objectives as a measure of planning effectiveness have better success. Empirical evaluation can be done by comparing a new method with a competing method, or by studying the impact of method instantiation on the environment and its users. In this case the *impact* is evaluated, because no comparable method was used.

I created and introduced the first draft of Usability Plan at late 1998. Since then, Usability Plan has had a lot of positive impact on the case project and parallel projects. A common language and practice has been established for planning the usability work company-wide. As a result of the experiments in this study, the Usability Plan has been adopted as part of the standard project planning documentation in the company, and it is actively used as reference when usability related issues are coordinated and performed. During the past two years a special training about usability planning has been available for designers and managers, and almost 100 persons have participated in the training this far.

To understand the benefits of Usability Plan I made an email questionnaire and interviewed 16 persons who have used the Usability Plan or who's work has been influenced by the Usability Plan in the case project and in the test projects (Figure 1.2): four technology and project managers, one usability manager, five user interface designers, three design engineers and three usability specialists at M3 milestone in the case project. The questionnaire covered the following topics:

- Have you used usability plan? What is the project phase/milestone currently?
- What benefits the planning has given to you?
- What problems you have encountered with planned (or unplanned) usability engineering?
- Is your (project) management committed to support usability work?
- Other experiences, preferences or improvement ideas?

The benefits emerging in interviews are, for example:
- less “random” design decisions
- less misunderstandings between people (designers - management)
- increased communication between people (designers - management)
- better defined usability studies
- better buy-in of usability activities and results
- project management has better visibility to the product entity, good understanding of the expected quality of use, and information for project planning and resource needs.
- cost-efficiency is good because the usability activities can be performed in a proactive way instead of reacting to observed problems, and because the resource need and use can be pre-planned.
- design teams have been able to plan their own efforts, especially in the design areas, in which it is necessary to have support from usability engineering.

The usability engineer has been able to plan the required work amounts, to have usability engineering budgets approved, and to organise subcontracting activities well in advance. The Usability Plan has had major impact on usability engineer’s long-term and short-term work. It has provided a way for design engineers to make sure they will get the necessary design support at the right time.

The project management has used the Usability Plan as a tool to verify that the issues important for the product success will also provide quality of use. However, the limitations mentioned by Daly-Jones et al. (1999) were also seen:
- usability plans were not stable because of changing project plans and development schedules.
- even though usability activities are planned, there is constant tendency to deliver some of the results too late in order to be effective on the design work.

A fundamental problem usability professionals report (Trenner and Bava 1998) is the lack of management support and commitment. By having the usability assessment meeting in the beginning of a project and an Usability Plan approved by the project management, the management has committed itself to support project usability engineering in a way that provides a solid platform for the usability engineer to successfully perform and overcome political problems.

The feasibility and effectiveness of the presented method can be empirically evaluated by assessing the outcome of the method instantiation (a project), i.e. the final products (March and Smith 1995). However, this evaluation is not in the scope of this study.

7.8 Summary

The process of designing and verifying the usability of a product needs well-planned and organised actions. Human-centred design and usability engineering are thus severe action points also for the management of the product development team. To deal with complex communication environments, technologies and terminals there should be methods to cover full end-to-end interaction scenarios. This sometimes requires close co-operation of different parties (product design team, network operator and service provider).

The presented methods for usability planning and usability risk management provide a systematic way to analyse the project needs and to plan usability engineering activities through the project lifecycle. The methods are built by:

a) differentiating coordination and performance tasks
b) differentiating coordination tasks into project coordination and coordination of usability engineering.
c) differentiating coordination further into planning and follow-up tasks
d) maintaining usability risk management data.

Usability Plan provides a method for synchronising usability engineering with other CE activities and coordination tasks. In general, a Usability Plan supports the project work effectively.

This study confirms the observation of Premkumar and King (1994) that a longer planning horizon enables better planning and performance. In this case the longer planning horizon is the product development lifecycle instead of a shorter timescale, such as a single usability test. The need for the top management's continued commitment during implementation of the plan is important, and according to my practical experiences in the case project, a fundamental necessity. It improves the Usability Plan for usability engineering Lifecycle (Mayhew 1999, Daly-Jones et al. 1999) by describing the planning tools in the framework of a CE project. The shortcoming of earlier descriptions is from the CE point of view, that they do not explicitly describe how the product design plans and states can be utilised for planning the usability efforts and how the project management can utilise information obtained from usability engineering for planning other project activities.

The finding of Blackburn et al. (1996) was that the concurrency is less prevalent between than inside design stages. We have shown (for example Table 7.2) that the usability engineering and the human-centred design activities can help in overcoming this discontinuation, especially in the turning point from the early design to the late design. Thus, concurrency of usability engineering is very much prevalent between stages and, unlike in software engineering, the concurrency of usability engineering increases while moving across software-hardware interface or across the stages.

The limitation of this approach is, that it requires early and continuous presence in the project to be efficient and effective, i.e. it encourages the project usability engineering instead of consulting-based or subcontracted usability engineering. There are two main lessons we have learned during this study:

- a project should have a Usability Plan as part of the project documentation in order to enable efficient and effective usability engineering.
- during the project usability engineering it is important to collect measurable data about performed actions, design improvements etc. in order to justify and better plan forthcoming usability engineering projects.

Is it possible to develop a method for planning usability work with CE? We have now shown that this is possible by adopting methods and practices used in other engineering areas and planning the actions according to CE process phases. However, the main consequence of concurrency for planning the usability work is related to organising the tasks. In a CE project the Usability Plan verifies that the project has capability to respond to the usability engineering needs of *concurrent* design activities.

The practitioners should strive for organising (coordinate and perform) usability engineering in the scope of the product development project instead of working in fire-fighting mode, using the same coordination tools as are used in other areas of the project coordination.

The goal of usability engineering is to ensure the quality of a system in terms of usability. The Usability Plan is a tool for enabling this. Further research is needed to define perceptual metrics for evaluating feasibility, accuracy and flexibility of a Usability Plan and usability risk management.

8 Managing Usability Engineering in Complex Concurrent Product Development

How can usability engineering be *managed* in highly complex innovative product development? When usability engineering is performed in CE product development, there are stages where usability engineering needs to be refocused in order to perform successfully in the changing project environment. The focusing points follow the product development milestones but are not identical with those. From usability engineering perspective those points are critical for achieving effectiveness and efficiency in the product development.

The objective of this chapter is to find out what kinds of usability improvements are possible in different smart phone development phases and how product development stages actuate usability engineering. This research problem has been addressed in Keinonen et al. (1996). Our approach to usability engineering is holistic, i.e. it views the product usability as an entity that is built from the user interface, the external interface and the service interface (see Section 5.2.1).

Research reports describe the practical usability engineering problems in product development and solutions for those problems. The two basic problems are:

- usability engineering is done too late and
- lack of management support for usability engineering.

If the development organisation is not well adapted to human-centred design these problems are likely to appear. In addition to these problems, in fast paced CE it is simply difficult to usability engineer all needed parallel design and engineering areas. Concurrent product development is a complex engineering environment.

Current understanding and description of sequential product development phases does not give much support for serious usability engineering in a CE project. CE does not match with human-centred design. Hakiel (1997b) emphasises the need for product engineering across disciplines rather than software engineering. This raises a practical problem in concurrent product development: If the product is complex and resources limited, what usability activities should be performed and when?

Though the usability engineering lifecycle (Nielsen 1993, Mayhew 1999, ISO 13407) is well defined and known, it is often difficult to apply human-centred design in concurrent product development lifecycle due to the fast development pace, complexities and because the product development is not fundamentally based on human-centred approach. However, the more complex the product is technically or conceptually, the more important it is to involve elements from human-centred design (Keinonen et al. 1996).

8.1 Approach

In this chapter I shall study especially:

- what are the critical points for usability engineering in a concurrent product development?
- how should usability engineering be refocused in those points in order to perform effective and efficient design and usability actions?
- what is the difference (turning point) when moving from early to late design in CE product development?

Concurrent product development sets both limitations and opportunities for usability engineering. During the projects it became clear that there are *predictable* turning points in product development where the usability engineering needs to be refocused. I call those *usability engineering milestones*. In addition, concurrency gives the perspective that there is no single early and late design phase, but one product development lifecycle contains at least two early and late phases in different engineering areas.

8.2 Concurrent Engineering Dependencies

In mobile phone product development there are several practical dependencies between engineering areas.

- product requirements, product concept, technical product platform and industrial design define the initial development frames for all engineering areas.
- industrial design defines the overall dimensions of the product and the main factors for mechanical design. It also defines the initial layouts of elements that are visible to a user.
- mechanical design defines the details of position and size of all product components and the available dimensions for hardware.
- hardware defines main performance issues, such as display capabilities, memory size and processor efficiency. Hardware enables certain software performance.
- software defines the software-based user interface capabilities.

The above-mentioned engineering areas have a final effect on user interface and so are potential subjects for usability engineering. From a user perspective, the product entity should provide quality of use. This implies that usability engineering should be done in all areas that have an effect on quality of use and usability. In a mobile phone, the areas extend from software to, for example, hardware, mechanics, ergonomics, out-of-box readiness, even network services.

8.3 Usability activities in product development phases

Following the product development phases the already known project milestones with usability activities are:

- M0: Start of usability engineering. The milestone is typically followed (or preceded) by requirements analysis and product specification.
- M1: Start of product design. This milestone starts (or continues) early design and formative usability evaluation. Usability engineering is done with low fidelity prototypes. The units for measuring effectiveness and efficiency are the number of identified design improvements and the capability to propagate the improvements to product design.
- M2: Start of detailed design and implementation. Usability engineering is performed with high fidelity prototypes.
- M3: Start of summative usability evaluation. This milestone starts usability evaluation in order to produce summative data about the product. The measuring units for effectiveness and efficiency are the number of found usability problems and the usability of the product.
- M4: Start of field feedback
- M5: End of usability engineering for the product. In most cases in the scope of this study (Figure 1.2), product usability engineering was ended before M4.

A notable observation is that the dimensions for measuring efficiency and effectiveness are different in M1 (capability to propagate the improvements to product design) and M3 (usability of the final product). The milestones (M0 to M5) and their position in product development phases are presented in Table 8.1.

M0		M1		M2		M3		M4		M5
	Specify product		Design		Implement & integrate		Test		Launch	

Table 8.1. Milestones and development phases.

Usability engineering milestones are product development stages where usability engineering needs to be refocused in order to provide efficiency and effectiveness in the changing design environment. A milestone is identified, for example, from the following characteristics:

- the project priorities or goals are defined or changed.
- a usability activity starts or ends (new usability activity starts, old activity is changed or ended).
- formal design support changes to summative design evaluation.
- target setting changes to planning, planning changes to follow-up.
- project practices for coordinating design changes is changed.
- usability engineering tool changes (for example from simulation to product prototype).

8.4 New usability milestones via horizontal and vertical review

Concurrent product development increases the number of previously introduced usability engineering milestones (M0-M5). New milestones can be found by following a process from start to end, identifying changes in the environment. Moreover, stopping at a specific place and looking at the activities that take place at different directions can also reveal new milestones. In the following I will study the CE process from two perspectives, horizontally and vertically, in order to identify new milestones.

8.4.1 Horizontal review

By horizontal project review (lifecycle) it is possible to identify the actual product development phases and their characteristics, and to estimate the potential effectiveness of usability engineering. Figure 7.2 shows how many design *changes* were made in different design phases and design areas in the case project. The findings based on a horizontal project view are:

1. Most product design (industrial, user interface, mechanical, software and hardware design) changes are made between M0 and M2.
2. Some changes are made even in very late design phases (software, localisation, user documentation).
3. Near M2 the product concept freezes. In this milestone the development team *exactly* knows for the first time what the product should be like. Before concept freezing, the organisation is *defining* the product, i.e. there are design options. After concept freezing the organisation focuses on implementing the concept and there are minimal design options. The product requirements are defined much earlier (M0), but in order to design and implement new functions or technologies for the first time it is necessary to leave the possibility for design changes to late design (Hakiel 1997b). This increases design uncertainty and design teams must be alert for implementing even surprising design needs.
4. The turning point from early design (creating the design) to late design (design changes and improvements) can be estimated. The turning point can be seen at M2. In the observed projects, this was the point when the development organisation started to use a design change management control mechanism.
5. Supporting engineering areas (localisation, user documentation) have early and late design phases.

6. Two freezing patterns can be detected. First, the main engineering areas (software, mechanics, hardware) freeze gradually from M0 to M2. Secondly, supporting engineering areas have a main design phase between M2 and M3 with fast freezing speed. Hence, product development has two separate phases: product design and support material design.

From this horizontal review the following new milestones can be proposed:
- M (EarlyToLate): Early design changes to late design.
- M (ConceptFreeze): Concept freezes. This gives us two important conceptual phases:
- Pre-concept-freeze design
- Post-concept-freeze design
- M (EarlyToLateSupportMaterial): Early design of user support material and localisation changes to late design of user support material and localisation.

8.4.2 Vertical review

By studying a project from a vertical perspective (concurrency) it is possible to identify the concurrency in product development phases, the effects of concurrency, and the potential usability engineering efficiency. The vertical findings based on Figure 7.2 are:
1. Development concurrency is real. For example, user interface design is concurrent with software design. 1. User interface design is aligned with the software design instead of being a preceding action (compare with Figure 2.1).
2. Designs freeze in a predictable order (1. concept, 2. hard design, 3. soft design, 4. Supporting designs).
3. Design changes are accepted in all design areas until M2.

8.4.3 Definition of early and late design phases

Product development can be divided into early and late design. The early design phase is characterised by interaction design and formative usability evaluations. Late design is characterised by detailed design and summative usability evaluations. Kiljander (1999), Nielsen (1994) and Mayhew (1999) emphasise the importance of focusing on early design phases instead of late phases due to high costs of changes in late phases. The estimate of increasing design change costs is based on findings in software engineering.

Earlier studies identify early design and late design as separate design phases. However, there is no definition of how a practitioner can identify the turning point from early to late design. Based on the observations above, I define early and late design phases for CE project, and the milestone between these phases, as follows:

- **early design phase** is identified by high capability and interest of an organisation to create new design, and accept and implement design changes to existing design. In the early design phase the target of product development is to maximise the number of design improvements in the given timeframe.
- **early design phase changes to late design** when the organisation *decides* to apply a (systematic) method for handling design change proposals.

- **late design phase** is identified by low capability and interest of an organisation to create new design, and accept and implement design changes to existing design. In the late phase the product development target is to minimise the number of design changes in order to manage project timetables and risks.

The ultimate goal of usability engineering is to support the project in achieving the project goals of cost, quality and time-to-market. The distinction and implication of early and late design phases are important: maximise innovation, design efficiency, effectiveness and quality before M(EarlyToLate) and minimise design changes after M(EarlyToLate). Usability engineering should work for common project goals and not against, for example in the name of best design.

8.5 Product launch and confidentiality

A mobile phone project is always highly confidential until the product is launched, i.e. the new product is made public. Depending on the company strategies, product is launched either during product development or when it is ready for market place. Confidentiality is a security risk for the development team and a challenge for human-centred design. By involving users from outside the organisation there is a risk that the product confidentiality is harmed.

Product launch is an important milestone for usability engineering. Before product launch user recruitment and mobile usability testing in a realistic context can be complicated or limited. After product launch user recruitment is easier and usability testing can be performed in public places, in the real context.

8.6 Milestones and simulations

The primary usability engineering tool is a product simulation or prototype. Depending on the product development phase different simulations and prototypes are available, low-fidelity, high-fidelity or product prototypes.

The product entity is composed from the engineering results in the late development phase (Figure 3.1). When a new product is developed, there is not a working product prototype before integration phase. The implication is that usability engineering on the product *entity* is not possible in the early phase without the aid of hardware or software simulation. Meanwhile, usability engineering is possible in the non-integrated engineering areas. In typical mobile phone development phases, the following prototypes can be produced:

- early design: low-fidelity (& high-fidelity) prototypes, design mock-ups, mechanics simulations
- late design: (low-fidelity and) high-fidelity prototypes, hardware prototypes, mechanics prototypes, partially working software
- integration phase: partially working product prototype
- testing phase: fully working product prototype.

The availability of prototypes and prototype functionality/maturity are major factors for usability engineering. Based on the prototype availability we can set prototype-based milestones (Mp) accordingly:

- Mp1: Low-fidelity prototypes and mock-ups possible.
- Mp2: High-fidelity and mechanical prototypes possible.
- Mp3: Partially working product prototype available.
- Mp4: Fully working product prototype available.

An example of timing for prototype availability is given in Table 8.2. The table shows that prototype milestones are not the same as product development milestones.

M0		M1		M2		M3		M4		M5
	Specify		Design		Implement & integrate		Test		Launch	
	Mp1		**Mp2**		**Mp3**		**Mp4**			

Table 8.2. Prototype milestones and development phases.

8.6.1 Research about prototypes simulations

Usability engineering, as commonly practised, is by nature based on the use of different kind of prototypes. Nielsen (1993) describes the dimensions of prototyping: vertical and horizontal. He notes that by selecting the dimension, it is possible to reduce the proportion of the system that is implemented and provide faster prototype production. A horizontal prototype simulates the functions but eliminates depth of functionality. A vertical prototype simulates full functionality of selected functions. Especially in the case of low-fidelity prototypes this distinction is an important choice to do because each user interface activity, for example managing months in Calendar application, is made manually to the prototype material, and also because the complexity of managing the prototype increases while the size of material increases.

Isensee and Rudd (1996) categorise prototypes to low-fidelity and high-fidelity prototypes. Low-fidelity prototypes are limited-function and limited interaction prototypes. High-fidelity prototypes are fully interactive and represent almost real user interface functionality. Several benefits for using low-fidelity prototypes in mobile phone design and evaluation are described by Kiljander (1997). Virzi et al. (1996) compare the usability problem identification with low-fidelity and high-fidelity prototypes. They report that the efficiency of low-fidelity prototypes, compared to high-fidelity prototypes, in finding usability problems is good throughout the whole product development cycle and that the number and type of identified problems are equal. They state that low-fidelity prototypes are an efficient way to search the design space, are predictive of preferences in the actual product, enhance user participation in the design process, enable visualisation of possible design solutions and provoke innovation.

Hansen (1997) reports on using paper prototyping for product design with very limited time and resources. She concludes with an observation that the packet of created paper prototype screen copies together with functional specification can work as product user interface specification. Kiljander (1999) describes experiences of using different user interface prototyping methods and tools in designing user interfaces for mobile phones. He concludes that low-fidelity methods are efficient tools to support rapid design iteration and to answer major design question, such as use of metaphors, content and navigation, in the early phases of the design process. Also Hall (2001) proposes the use of low-fidelity prototypes, especially when new technology is developed.

8.7 Paper prototypes as the main tool for early product development

In order to perform effective and efficient usability engineering in the early development phase, a product simulation is required. Thus, the development of simulation is an essential part of early product development and should be an integral part of product development process. Milestones Mp1 and Mp2 are hence more important milestones than Mp3 and Mp4.

Low-fidelity usability testing, especially with paper prototypes has a long and established tradition at NMP. It is efficient technique due to capability for fast design changes and iterations in the user interface design. However, the results of single tests may be unreliable or vague and techniques for improving the reliability of low-fidelity testing need to be developed.

Honold (1999) made an empirical research about differences in the way cultural groups use mobile phones and to determine what product adjustments were necessary to ensure culturally suitable design. She concludes with the finding that it is not enough just to translate scenarios of usage situations, because a lot of existing differences between culturally different user groups are levelled off due to the artificial similarity of the usability testing simulation. Although research about usability testing in different cultures is reported, there is no research about how well paper prototypes suit for usability testing with cultural factors and what are the potential advantages of combining these methods. In addition to that the applicability of using low-fidelity prototypes, especially paper prototyping, for product design and usability testing in target market areas in large scale is not well experimented or reported.

8.7.1 Analysis of the data and results

As part of this study, we made a series of four usability tests with paper prototypes (described in Article V). Our aim was to study how the reliability of usability tests can be increased through repeating the tests, and how well the low-fidelity prototypes work in different cultural contexts. The four tests, each with 10 test participants, created a large number of insights, information about the acceptability of the new user interface style, pinpointed specific design problems and yielded several new design and localisation ideas. Test also gave us information about *how* some new functions may be used and how they are understood. As expected, users were not paying attention to draft graphics and incomplete screen designs. The types of findings are in Chart 8.1.

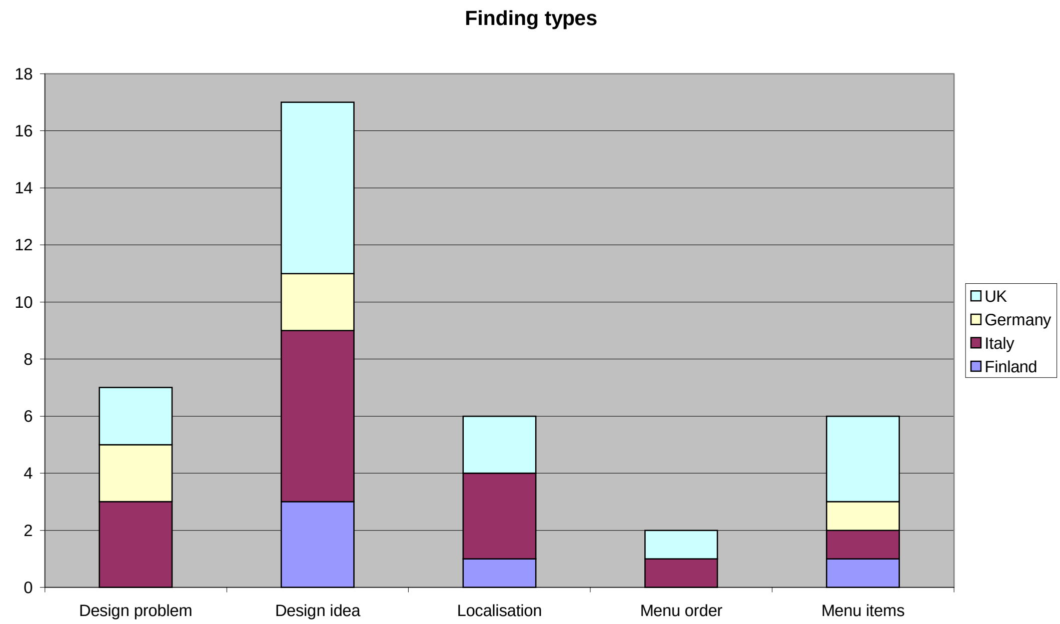

Chart 8.1. Findings and types in different tests.

The findings confirmed, contradicted or were neutral with other findings in this study. Several findings were recurring observations, only few being contradictory, and they were seen in other test tasks and test sessions. These were typically UI design problems. New UI design ideas, most of the findings, were created based on user actions or as a consequence of designer's insight. These ideas were not directly correlating with other findings, but they could be assessed in other test sessions. The contradictory findings were typically user preferences about the order of menu items or comments about localisation.

Dumash and Redish (1993) present a classification of global usability problems found in software and list the typical problem issues. The following table (Table 8.3) lists the main problems that we found and a mapping to corresponding items in Dumash and Redish classification. Leftmost column (Observed problem) lists the finding types in our study. Rightmost column (Dumash and Redish) lists the finding types identified in their work. The classification of Dumash and Redish captures half of problem types that we also observed during the four tests (white cells with text). The types that were new (grey cells in the table) are: missing shortcuts and menu items, the not-optimal order of menu items, text proposals, finding a function and interaction problems. The table also shows that most of the observed were such that they could be fixed to final design, in other words, the efficiency of usability engineering was good. Most of the observed usability problems were such that they were observed at least in two countries. For example, a specific but not evident menu command that was not in the prototype was requested by a user in every country. This kind of observations can be handled as reliable findings and as severe usability problems that need to be fixed.

Observed problem	**Fi**	**It**	**Ge**	**UK**	**Fixed**	**Dumash and Redish**
Option list is too long	X	X	X	X	X	Too many menu options
Requested menu option is missing	X	X	X	X	X	
The order of menu items is wrong				X	X	
Numbered menu items requested		X			-	
Don't understand that there are more options than shown on screen	X	X	X	X	X	Don't understand that there are more options
Colors should link to user interface and physical buttons better	X	X			-	Function keys arbitrarily mapped to functions
Immediate editing should be possible in certain kinds of forms		X	X		X	Waste time entering the data
A note/expression/term is not understood		X			X	Error messages need additional level of detail
A confirmation note is missing				X	X	No confirmation that user intends action
An information note is missing				X	X	No message that action has been taken
Shortcuts for function is missing (several places)	X	X	X	X	X	
Term not understood		X			X	Menu options use jargon
Term localisation proposals	X	X	X	X	X	
Functionality is available too deep in menu structure		X	X	X	X	
Handling of data connection termination is not as excepted			X	X	X	
Default functionality of selection key was not as expected				X	-	

Table 8.3. Usability test findings in our study, and their correlation with findings of Dumash and Redish (1993).

The cultures in the selected four countries are not very different. They belong to the same European culture area and have many common issues in cultural, religious and historical background. In spite of the cultural proximity, the tests in different countries gave slightly different results. Some of the problems were seen only in one country (but in some cases by several users) and they were most often reflecting a missing note or a text that was not understood. The reliability or unreliability can be handled by iterating and repeating the tests. These observations require either language specific fixes (localisation) or simple design fixes for adding feedback and information. There were also several culture-specific observations. They were not recorded as user interface design proposals, but they were recorded as "insights".

8.7.2 Applicability of paper prototypes for mobile phone development

The main benefits of paper prototypes for mobile phone development are following:

- paper prototypes enable reliable user testing of interaction in the early development phases.
- the form and size factors of mobile phones are optimal for paper prototypes
- low-fidelity (paper, cardboard) mock-ups are not good for testing industrial design if they are produced manually.
- user testing with paper prototypes is efficient in producing new design ideas.

One of the strengths in low-fidelity prototyping is the ability to easily simulate situations that are not yet more than ideas in the designer's head, or that need instant user testing without time or resources to produce a computer-based simulation. This is an opportunity for early product design. With paper prototypes it is possible to overcome some of the challenges related to the late integration of the product, for example, the actual product prototype is working only in the late phase of the development. Expressed in a more concrete way, with paper prototypes we have been able to user test the phone functionality even several months before its first versions have been available in the real software.

Paper prototypes are ideal low-fidelity tool for designing and testing mobile phones *user interaction* for several practical reasons, the main benefit being that the user is able concentrate on the essential interaction instead of being misled by unfinished or wrong user interface details. Paper prototypes are mobile in the very same way as mobile phones are. Paper prototypes allow real in-context design and following of the user in mobile activities, for example train or airplane. Typical screen size of mobile phones (~3cm*~4cm) suggests the use of small paper prints, standard post-it notes, because entire screens can be easily and fast sketched to one note. The use of paper prototype supports taking the form-factor better into account than, for example, in testing with a laptop computer and user interface simulation. Moreover, paper prototyping can't simulate dynamic communication situations very well, for example a voice call interaction. Audio feedback (recipient voice, voice quality, notification sounds, etc.) is very important in mobile phone interaction and the lack of this information may decrease the reliability of those test results that are related to voice communication.

To some extent, it is possible to test industrial design with low-fidelity prototypes, for example, as a cardboard mock-up (for example, Säde et al. 1997). This may give useful information when the ergonomics or form-factor is important. However, the cardboard mock-up is easily seen as real presentation of the phone's real size or shape, which are often too difficult to correctly produce with simple manual prototyping techniques.

The number of new design ideas is considerable (Table 8.3). If the paper prototype test objective is to capture new design ideas and evaluate proposals made by the test user, this method allows instant design changes and thus immediate feedback to the user's comments. In this study the most instant design changes were terminology updates and adding of information notes according to user comments. This method makes it possible that other members of the test team are able to produce new design material during the test session and even during a test task, supporting test situation in an interactive way. For mobile phone user interface design this possibility has been a major source of new innovations and for simplification of interaction sequences.

8.8 New milestones and job design

Milestones are important for job coordination. We have now seen that there are several possible new milestones where the changes in design environment have an effect on usability engineering (Figure 8.2). Those places are related to:

- General product development stages
- Capability to accept and implement design changes
- Capability to perform iterative design
- Current design environment, availability of prototypes.

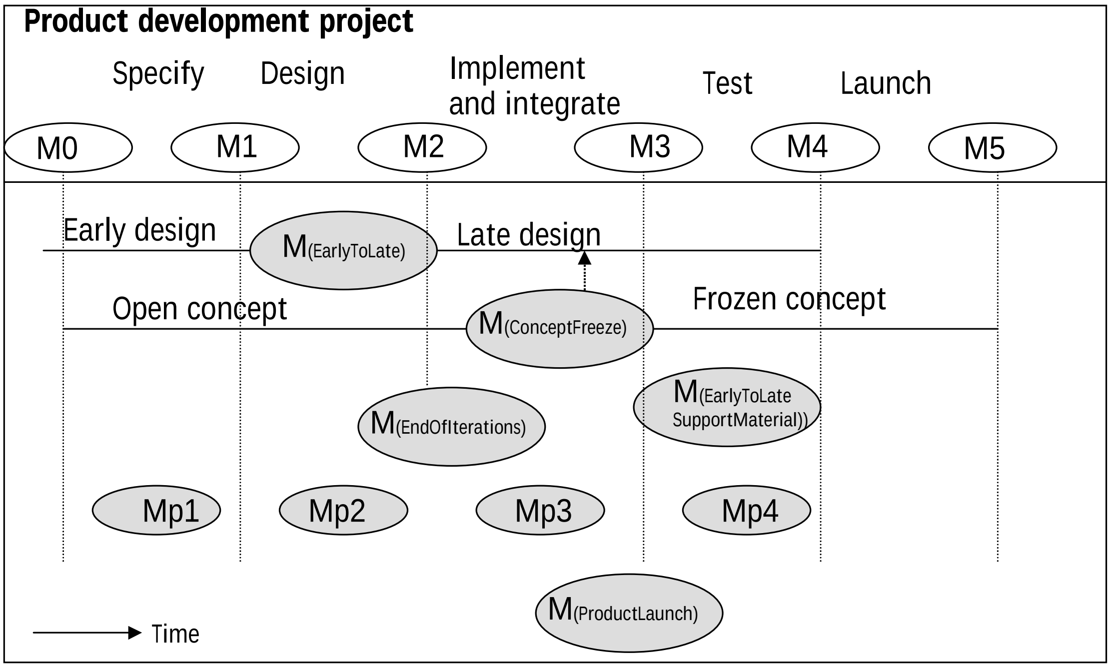

Figure 8.2. The effect of milestones during product development on usability engineering.

New usability milestones do not mean that we need more bureaucracy in the project. Knowledge and awareness about milestones and turning points can be used in planning the usability work and job design, in the same way as land marks, map and compass are needed for navigation. In the planning phase by identifying the opportunities such as available prototyping tools, and limitations such as project state, the usability practitioner can prepare to work in an optimal way, effectively and efficiently.

By taking milestones into consideration it is also possible to improve cost efficiency of usability engineering by focusing on areas where changes are possible and where usability engineering is mostly needed. New milestones have two practical consequences related to work organisation:

- new usability engineering control points and follow-up mechanisms may be needed.
- to some extent, job design or redesign is needed in each milestone. Earlier tasks can disappear and new ones arise (Järvinen 1980).

The identification and definition of milestone M(EarlyToLate) is especially useful. The spontaneous tendency of designers is to improve the designs as long as possible, while the will of project management is to minimise all changes after M(EarlyToLate). An explicit definition of this milestone provides improved understanding of the common goals and helps to organise the design work in a better way.

The positive effect of new milestones is the improved ability to coordinate usability engineering. The negative effect of new milestones and job design is the unproductive nature of tasks. Control and job re-organisation do not contribute to actual product development (Järvinen 1999).

Usability engineering could also be coordinated with fewer milestones. Removing milestones would potentially lead to improved capability in following the principles of human-centred design, especially iterative design. The drawbacks would be weaker capability to organise the work in fast product development, and weaker visibility and linkage to the product development entity.

8.9 Summary

The challenge of usability engineering in concurrent product development is to verify that the final product is usable and that sufficient usability engineering is done in all design areas that have an effect on user experience. We need efficient but simple ways to integrate usability engineering activities into the concurrent development process following the principles of human-centred design.

In this chapter, I have introduced a new concept (usability milestone) that can be useful in concurrent usability engineering. Based on this new concept, I have identified and analysed several new usability milestones and discussed their effect on usability engineering. The main result of this analysis is that usability engineering *can* be successfully performed in fast concurrent product development but it may require better control mechanisms than described by current research. The control mechanisms can be based on the identified new milestones.

CE projects are time-critical. The attitude towards usability activities is often that they delay product development and cause extra work (Kaderbhai 1998). An important objective for usability engineering is to support the project in achieving the milestones in the planned timetable. This can be achieved when usability engineering is an integral part of the design process and adapted to the project limitations and opportunities.

A common problem in concurrent product development is timetable delays. By decreasing the design phase time and increasing the design quality and efficiency with usability engineering, it is possible to provide either more optimal or more reliable timetables and time-to-market. Moreover, usability engineering typically requires time-consuming iterative design.

Early phase front-loading in the form of early user testing in product development and evaluation can be used to foresee user problems and to decrease usability risks. When low-fidelity prototypes are used with early design specifications in user testing, there is a higher probability of producing unreliable results. The reliability can be considerably increased by repeating the tests, either as part of design iteration or by usability testing the same design more than once.

Early and continuous user-involvement in the product development, as proposed by human-centred design processes, can be problematic due to confidentiality requirements. Very often this leads to the situation where the design is tested mostly with persons that work inside the organisation, i.e. the *real users* do not participate in the design process, or the testing is done without real context. There is a need to develop usability testing methods, especially for mobile context, that combine early development needs, context of use and confidentiality requirements.

The introduced usability engineering milestones may

- be useful and applicable for any sequential development process that is based on milestones and phases, for example the Concurrent Engineering process or the hierarchical spiral model (Iivari 1987), and where the outcome of the process is a concrete physical product, such as a mobile phone
- be only partially applicable to product development practices where the process is not based on milestones and phases, and where the outcome of the process is not a concrete or independent product, such as an improved software component.
- not be useful for short-term and simple development projects.

By analysing potential usability milestones in the beginning of a project it is possible to make a usability plan enabling efficient and effective usability engineering within the given project targets and timetables. New usability milestones are useful if they improve or verify the quality of the product. The efficiency and usefulness of usability engineering milestones can be measured, for example, with effort metrics (Höglund 1999): time, cost and results. This kind of measurement is possible when there is reference data from other projects.

Based on the findings in this study, I propose questions for further study. Can usability engineering decrease the uncertainty or delays in product development? How applicable and useful is the M(EarlyToLate) milestone? Are the findings of this study applicable in other industries and in other models of product development, for example Waterfall method-based software engineering? And finally, can we develop better usability engineering practices for CE using better-defined usability milestones and knowledge about the turning points?

9 Guidelines for Integration of Concurrent Engineering and Usability Engineering

The previous chapters have shown some usability engineering experiments and experiences in a case project. The experiences as such are interesting, but further development is needed to turn the experiences into guidelines that can be studied by researchers and applied by practitioners.

One of the main learning this far is that usability engineering lifecycles (Table 2.1) are not enough for concurrent product development because they lack the aspect of coordination and complexity. Also, practical difficulties to assure product usability, such as availability of prototypes, are often underestimated. This chapter is an attempt to formulate the experiences as guidelines that promote human-centred design, and which can be used in concurrent consumer product development, evaluate the proposed guidelines and methods that were built and used in the case project, and to provide evidence that proposed methods are more efficient and effective than competing approaches.

Examples 2 (Wall Street Journal 1999) and 3 (Business week 2001) show two cases where the usability engineering has not been successful or sufficient. Looking from insider point of view (researcher is part of the development organisation) it is possible to identify some of the reasons that led to insufficient usability engineering:

- organisation and process: The authority of usability engineer was not clear. The usability engineering was not systematic, planned or continuous throughout product development lifecycle.

- product structure: Focus of usability engineering was strongly on the software functions. Product entity, accessories, manuals and services were not on the focus of UE. It must be noted that the examples 2 and 3 do not address software functions nor user interface problems, but rather issues considering external interface, service interface and the physical product.

- user requirements: The example products (Nokia 9110 and Nokia 9210) have extensive wireless Internet connectivity capabilities (for example, Email, WWW, WAP and MMS). The major problems shown in the articles are related to the set-up of Internet connectivity. My conclusion for explaining the problems is that requirements related to Internet set-up process have not been known or understood. The articles also show that without fundamental understanding of these requirements and responding to those, the same problems will be also seen in further products.

In the product development, some of the usability problems (1) are identified during the product development and (2) some of the problems appear only after product or prototype is in use. From those problems identified already during product development some are corrected, but part of the problems remain problems also in the final product, for example due to timetables, implementation risk minimizing in the late phases, or simply due to task priorisation. The overall challenge for usability engineering is to minimise the portion of (2) and maximise the portion of (1).

According to Göransson (2001), very few companies and organisations are willing to go the whole way and adopt a truly human-centred design process. These companies and organisations realise that being committed to usability means that there must be a shift in the way that they develop interactive systems. By defining the results of this study as guidelines, rather than well-defined process model, we provide flexibility that is needed when applying the guidelines in different kind of development organisations. The guidelines make it possible to introduce human-centred design elements in the development without disruption, with small steps that can be employed. This approach also follows the idea of CE being a management tool (Blackburn et al. 1996) or collection of methods, support tools and ways on how the development is organised (Heikkinen 1997, 10). The usability engineering guidelines belong to the Engineering Support Initiatives (Human Factors) category of CE components.

In this chapter I try to provide a practical answer to the initial research question of this case study: how can Usability engineering be performed effectively and efficiently in a CE mobile phone development project? Our findings show that by accepting the limitations and development complexity that CE sets for usability engineering, it is possible to adapt usability engineering to enable efficient and effective activities. The findings also give evidence that with the proposed activities, in particular Usability assessment meeting and Usability planning, it is possible to overcome the most common organisational problems identified by usability engineers.

The rest of the chapter is organised as follows. In the next Section (9.1) I shall discuss the real life usability engineering tasks performed in the case study. Organisational setup of a usability engineer is discussed in Section 9.2. Section 9.3 organises the experiences as guidelines. Section 9.4 evaluates the success of case project and presents evidence about the efficiency of proposed guidelines.

9.1 Usability engineering tasks - some examples

During the case project usability engineering has practically covered the whole product design with approximately 50 user tests and more than 300 individual users in several development sites of the multi-site -organisation. In order to understand the diverse usability work field, some examples are now described.

Mechanical product concept and industrial design. The product concept (Figure 1.1) is based on moving cover, large colour display, some new function buttons and a 5-way joystick as navigation tool. For example, the basic concept idea of having a joystick instead of traditional navigation buttons has needed concept testing and design iteration.

User interface design. The user interface introduces a new UI style (Nokia series 60) and large number of novel functions, such as multimedia messaging and Bluetooth connectivity. Continuous (iterative and repeating) user testing and designer co-operation, starting from M1 with paper prototypes, has been performed in the design areas and user tasks that were considered as important for the success of the product and user acceptance. Due to the diversity of the product, all details of the product design have not been possible to user test.

Mechanical design. Human-centred development of mechanical design covered all the areas where user needs to touch or hold the product, including tasks such as SIM card insertion. This area is especially challenging because the product's touch-and-feel is highly dependent on the quality and materials of the prototype. For example, the joystick design has required large-scale iterative user testing in order to find the correct dimensions, material and the right "touch". Also, in order to meet the required product quality and robustness mechanical design has been supported with user-originated knowledge, such as, how many times a specific function is used during the product lifetime.

Software design. Even though user interface design deals with SW design, there are areas that require human-centred design but are not part of design of the visible UI. For example, there are several system time-outs, memory limitations, task durations, and invisible functions correcting user errors that have been adjusted based on the knowledge of usage patterns and user testing.

Hardware design. Hardware defines main performance issues, such as display capabilities, memory size and processor efficiency. Hardware also covers issues *visible on the UI*, such as backlights (brightness, colour) of the display. Field and laboratory testing and iterative design have been performed in order to meet user requirements of the backlight quality.

Localisation. All user tests have produced information needed for product localisation, and also found logical and syntactical inconsistencies in the user interface terminology.

User documentation (manuals). User documentation should help the user in overcoming emerging problem places in product use, and also provide useful information of the product in general. Continuous user testing has helped the creation of user documentation mainly by pinpointing the context-dependent problems places in user interaction, and further in iteratively testing the user documentation.

External interface design. The examples 2 and 3 address problems related to the service interface, product configuration and first use in particular. The shortcomings of earlier and competitor products have been analysed, and the information has been used in designing for the first use. Service interface and configuration has been an object of intense human-centred development in UI design, SW design and service design areas.

9.2 Organisational issues

The organisational position of usability engineer in the case project (Figure 1.3) was under project management but outside concurrent design teams. Along different phases of the project this turned to be a fruitful situation in three ways. First, the project management had a single usability expertise "access point" available for evaluating management level decisions about product functions from user point of view independently of the domain area. Secondly and following the first note, the concurrent development teams had a single and *neutral* "access point" for usability related design decisions. The development teams could always have an evaluation of design issues independently of team-internal interests, conflicts and preferences. Thirdly, the usability engineer could evaluate and involve in any area of product design. By contrast, for example, if the usability engineer is part of software team, he is not (typically) allowed or authorized to evaluate, for example, mechanical design. In addition, by having view and access to all design teams the usability engineer could have an early understanding of product entity and share this understanding to teams with limited understanding.

The most common and severe usability engineering problems were listed in the beginning of this study (Chapter 1):

- *organisational and managerial problems*, such as lack of management support and understanding the profits of usability engineering
- the *cost of design changes* in the design process increases rapidly when the design moves from the early phases to the later refinement and testing phases
- the problem of *usability testing too late*, causing high-cost or unimplementable changes.

One of the fundamental results of this study is that these problems can be eliminated or considerably decreased with proper organisational position of usability engineer, by involving project management through usability assessment meeting and by planning the usability work to cover whole development lifecycle keeping the product entity and product usability risks in mind. Detailed evidence of these issues is given later in this chapter.

9.3 Usability Engineering Guidelines

Lessons-learned in the case project and other observed projects can be expressed as engineering guidelines. The guidelines are structured following Tianfield's (2001) discussion about product development complexity. By using word *guidelines* I emphasise that they are engineering ideas tested in practice, and may be useful to other usability engineers. The guidelines are not mandatory for ensuring success; it is up to the practitioner to decide which, how and when guidelines are useful in the specific development setting.

9.3.1 Adjust Usability Engineering to Complex Organisation

The organisation of Nokia Mobile Phones is complex due to high number of employees, multi-site product development, cultural diversity, high degree of virtual teams and personal networks. Due to the Law of Requisite Hierarchy, the organisation continues growing even more complex because of the need to increase the product variety. Continuous flow of new products is an opportunity for product development. It enables quick learning from recent products that are on the market and available for field studies. Flow of products also makes it possible to share resources with parallel projects.

Usability engineering methods or lifecycle descriptions do not give guidance to the organisational aspect, because the focus is mostly on describing the optimal way to design usable products in a simple organisational setting. However, there are some sources giving examples of successfully managing usability engineering in also large companies. Wiklund (1994) provides an extensive overview over several U.S. companies (Kodak, Apple, Compaq, DEC, Hewlett-Packard, Silicon Graphics, Borland, Lotus, Microsoft), mostly focusing on the software industry. Radla and Young (2001) describe lessons learned from starting usability activities in three companies:

- excellent interpersonal skills are crucial to developing relationships with development teams.
- applying the results of HFE (human factors engineering) activities to thought leadership in product development makes the company more successful and raises HFE's credibility
- working directly with the customer creates high visibility with management, marketing, and product teams.
- even when schedule pressures are intense, HFE is possible.
- expensive labs and equipment are not necessarily prerequisite for HFE.
- most resistance comes from other pressures (such as schedule) and lack of information.
- there is no substitute for observing user interactions first hand

Guidelines to deal with complex organisation

1. The role and organisational position of a usability engineer needs to be clear, accepted and understood by other members in the development organisation (project).
2. A usability engineer needs to have authority to co-operate with different kind of teams in many product development sites. A usability engineer needs to co-operate and involve in the work of local and remote, natural and virtual product development teams.
3. The existence of multiple product development sites should be exploited, because they provide straightforward access to local cultures and users.

4. The continuous and parallel flow of products should be exploited in the form of continuous building of domain specific usability knowledge, in order to efficiently share product understanding and to share usability resources.
5. Formal training and education of designers should consider similarities to other design fields (Karat and Dayton 1995).
6. More members than just usability specialists in the development team needs to be involved in usability activities (Karat and Dayton 1995).

9.3.2 Adjust Usability Engineering to Product Development Process

The CE product development at Nokia Mobile Phones is well-defined process, with special emphasise in the timetable reliability and predictability, and product quality. CE process is applied by defining common milestones for parallel project activities. Parallel development activities are controlled and development risks minimised with the milestone reviews.

Milestones are useful for usability engineering because they provide predictable information about the expected advance of a project (timetable), and beforehand define what are the expected deliverables of different design areas in each milestone. Especially, the knowledge about early design phases is important because this is the most cost-efficient time for usability engineering in a project.

Guidelines to deal with product development process

7. Usability engineering should be aligned with product development milestones and processes. In each milestone (n), the knowledge from previous milestone (n-1) is the input data. Input data can be also achieved from other development projects. The next milestone (n+1) is the next target. The usability engineering goal for the next target should be formed based on the current (n) understanding of product strengths, weaknesses and identified design risks. The transformation (n-1 → n → n+1) needs to be known for each concurrent process where usability engineering is performed.
8. The defined deliverables of milestones should be used as input information for usability engineering.

9.3.3 Adjust Usability Engineering to Complex Product Structure

Mobile phones, especially smart and innovative products have often technically complex product structure. Product usability is dependent on a number of product elements in the user interface, the external interface and the service interface. During the development process the product components are designed in parallel and integrated in the late phase. The success of integration, including accessories and services is based on well-defined specifications and deliverables of different product development process. The product testing (system testing, integration testing) is an intensive and demanding task, and must be managed in smaller units during the product development. The complexity of product structure sets many implementation risks to product development difficult to manage. It also sets many risks to user acceptance. Examples 2 and 3 indicate that along the increase of product complexity, the need for product customer support increases.

The weaknesses in existing usability engineering process descriptions, for example ISO 13407, is that they are software oriented, are not well-adapted to mobile context (but rather office appliances and software), and they do not take into consideration product ergonomics and the three interfaces.

Guidelines to deal with complex product

9. Usability engineering should cover, in some form or another, all product elements that have an effect on usability.
10. Usability engineering needs proper coordination (planning and follow-up). Innovation and design efficiency should be maximised before M(EarlyToLate) and minimised after M(EarlyToLate).
11. Product risk management should be supported with the human-centred knowledge gained through user testing.

9.3.4 Adjust Usability Engineering to Complex User Requirements

When the new product introduces functions not familiar to users or designers, the designer often has not enough knowledge about the user requirements. Sometimes the users do not have specific needs and correspondingly requirements, towards the product or new functions, and the needs must be created by marketing activities. Especially with innovative functions, the elicitation of user requirements is a large job and may not be practically at the hands of a designer or usability engineer. In the worst case, without access to the users, the only method for identifying user needs is introspection (Järvinen 1999, 100). Due to individual differences and lack of real use context this potentially leads to design uncertainty.

Organisational memory is cumulative information about history having effect on the decision making of today. It consists of three components: the traditional knowledge bases, experts and prototypes. Information about existing solutions resides within the artefacts (prototypes) themselves (Hargadon and Sutton 1997). Walsh and Ungson (1991) conceptually analysed organisational memory, especially the three processes: acquisition, retention and retrieval. They identified five storages (bins) for organisational memory: individuals, culture, transformations, structures and ecology, but failed in capturing prototypes as storage. Organisational memory is perhaps the most powerful "tool" in dealing with complex and emerging user requirements. Usability engineers have an important role in building and structuring those parts of organisational memory that deal with detailed information about end-users' requirements, needs and skills.

New functions, especially in information technology products, are created by combining or restructuring existing technologies and functions, or by introducing a whole new technology or function. New functions are also often combined to larger service entities, end-to-end solutions. User requirements for this kind of functions are complex. According to Hargadon and Sutton, designers and usability engineers need to exploit their access to a broad range of technological solutions with organisational routines for acquiring and storing knowledge in the organisation's memory and, making analogies between current design problems and past solutions they have seen, retrieving that knowledge to generate new solutions to design problems in other industries.

Guidelines to deal with complex product requirements

12. Elicitation of user requirements should be done outside and before the product development because this part of the work is time-consuming and may not fit in the intense product development timetables.
13. At least introspection should be used for finding user requirements. However, this is recommended *only* when nothing else can be done in order to find out requirements.
14. Knowledge about user-requirements obtained from simulation or product prototype testing should be actively collected and shared in the organisation.

15. The design and usability testing for first use (out-of-box readiness) need to be included in the early product development.

9.4 Evaluation of the guidelines

The guidelines form a preliminary set of *normative instructions* to be used in future methods. The guidelines are intended to support meeting the needs of different organisations. The guidelines describe how usability engineering ought to be performed in order to be efficient and effective (March and Smith 1995). For normative guidelines it is not possible to say whether they are correct or incorrect, one can only evaluate whether they are useful or reasonable, and whether they improve something that already exists (March and Smith 1995, Järvinen 1999). According to March and Smith, evaluation of instantiations considers the efficiency and effectiveness of the artefact and its impacts on the environment and its users.

Usability engineering guidelines are useful for any development where the user acceptance and usability are important factors, and when the development:
- is done in a complex organisational setting (organisation, processes, product, requirements)
- takes a long time (>2 years)
- is concurrent (several parallel development teams)
- is done in multiple development sites
- deals with high degree of innovation and difficult-to-identify user requirements
- the product to be developed is an integrated entity of hardware and software.

Usability engineering guidelines may not be useful in a simple organisational setting where the development time is short and the product to be developed is conceptually simple, such as a stand-alone software application with limited number of functions.

Ulrich and Eppinger (1995) present characteristics of successful product development:
- product quality. How good is the product resulting from the development effort? Does it satisfy customer needs? Is it robust and reliable?
- product cost. What is the manufacturing cost of the product?
- development time. How quickly did the team complete the product development effort?
- development cost. How much did the firm have to spend to develop the product?
- development capability. Are the team and the firm better able to develop future products as a result of their experience with a product development project?

I apply these characteristics for evaluating the success of methods presented in earlier sections, and guidelines presented in this section. In particular, I shall discuss:
- do the proposed guidelines improve product quality from user point of view?
- do the proposed guidelines have an effect in product cost?
- do the proposed guidelines help in decreasing development time?
- do the proposed guidelines enable better cost-efficiency for usability engineering?
- do the proposed guidelines increase the ability to build more usable mobile phones?
- do the proposed guidelines increase efficient and effective usability engineering?
- in addition to product development, it is interesting to discuss whether the guidelines can be used to improve existing usability engineering methods or practices.

9.4.1 Product quality

Do the proposed guidelines improve product quality from user point of view? By applying Usability Plan it is possible to focus and prioritise usability engineering into issues that are most critical for user satisfaction, and to share the understanding of most important issues with project management and design teams. In my experience, Usability Plan is an efficient tool for developing a well-integrated product entity that meets user needs in the most important areas.[Guidelines 7, 8, 9, 10]

With Usability Risk Management the usability engineer can predict, keep track of and communicate the emerging design problems and product challenges that may cause the failure of product in terms of user acceptance. In the case project it was observed some predicted usability risks emerging, such as too high error rates in the user interface navigation. [Guideline 11]

9.4.2 Effect in product cost

Do the proposed guidelines have an effect in product cost? Product cost is a result of several factors, such as development time and resources, manufacturing, marketing and aftersales activities, and maintenance services. Decreasing any of these areas decreases product costs.

Smart products tend to have functions that are not used, for example, because they are too difficult, or they are not needed. Still, they may be required for marketing purposes, for example. This *creeping featurism* (Norman 1998, 80) is typical in the introduction of new technologies. Development of any function requires specification, design and implementation, i.e. manual work and resources. By early understanding of user needs, user requirements and user testing it is possible to avoid the implementation of unnecessary functions. [Guidelines 4, 12, 13, 14]

In the case project I could detect decreased project cost elements due to planned usability engineering in:

- product development in the form of better (faster) function and technology selection and design. For example, a complex software feature related to phone set-up was not implemented at all because of the negative user feedback in the early phase of specification.
- aftersales and maintenance services[13] in the form of better design of easy first use situation and product configuration. For example, new external interface concepts related to the messaging capabilities were studied and improved in parallel with the user-interface design.

9.4.3 Development time

Do the proposed guidelines help in decreasing development time? Product development time is the major motivation for applying CE. Development time in CE is dependent on the optimal implementation of milestones and success of product integration.

In the case project I was often in the situation when the design work in some design area couldn't proceed without an important selection between different design candidates, and any delay would have delayed the whole project due to design interdependencies. For example, the early selection of phone keypad design and material required user testing in order to find out which is the preferred way (design, technology) to proceed. The test need was foreseen in the Usability Plan and usability testing could be arranged without delays when it was needed for producing data. [Guidelines 7, 9, 10, 14]

[13] Lano and Haughton (1992) informed that the costs of maintenance activity have been estimated as being as high as 80% of the long-term cost of developing and maintaining systems; and this proportion is rising.

9.4.4 Cost-efficiency for usability engineering

Do the proposed guidelines enable better cost-efficiency for usability engineering? Design changes in the early development phases are cheaper (time, money, resources needed) than in the late phases. Fewer and smaller changes can be implemented in the late phases than in the early phases.

Compared to earlier development projects, more cost-efficient design changes and improvements could be made in the early development, and less change needs emerged in the late phases due to proper planning and priorisation of usability activities. However, in spite of better planning the product development still contained unexpected usability activities, but which could be managed due to proper planning in other tasks. [Guidelines 7, 10, 12, 15]

9.4.5 Usable mobile phones?

> *The high-tech industry is in denial of a simple fact that every person with a cell phone or a word processor can clearly see: Our computerized tools are too hard to use. The software engineers who create them have tried as hard as they can to make them easy to use and they have made some minor progress. They believe that their products are as easy to use as it is technically possible to make them. As engineers, their belief is in technology, and they have faith that only some new technology, like voice recognition or artificial intelligence, will improve the user's experience (Cooper 1999).*

Do the proposed guidelines increase the ability to build usable products? In my experience, by following the guidelines it is possible to better adapt usability engineering to support the development of complex products, and to better understand the particular product entity (the user interface, the external interface, the service interface) and the user requirements.

In the case study organisation (Nokia Mobile Phones) the improvement in product usability and usability engineering competence has been very clear. Usability engineering efficiency and effectivity is clearly better in the case project (and test projects 1-3) than in the observed projects (Communicator 1-3). Level of usability engineering ([Coverage, Duration, Visibility], see Section 6.3) has improved at least from [x,y,z] to [x+2, y+1. z+1]. The future of imaging phones will show, whether the products really are usable.

9.4.6 Efficient and effective usability engineering

Do the proposed guidelines increase efficient and effective usability engineering? The terms were defined as follows:

Efficient usability engineering is to work in an effective and competent manner in the current development phase with little wasted effort. Efficiency can be measured by comparing realization (output) against usability engineering goals, plans and amount of work (input).

Effective usability engineering aims to produce an adequate or desired result. Effectiveness can be measured by studying how usable the final product is, or by analysing field feedback. During product development the effectiveness can be measured, for example, by comparing the performed usability work against the number of usability engineering originated design improvements.

The usability engineering has been efficient in the case project and also in the followed three projects in terms of perceptual measures. It has been possible to execute Usability Plan and to perform the planned tasks. The current situation shows visible improvement to the situation 12 months earlier. Now (May 2002), at NMP many new development projects have a Usability Plan accepted by the management and implemented by usability engineers.

The usability engineering has been effective during the product development. Usability engineering originated design changes at early (before M2) development have been implemented with the rate of 60% (6 proposed changes out of 10 are implemented), and ~30% for late changes (after M2). The first product that has been developed using these guidelines is Nokia 7650. However, in order to assess the real success of guidelines it has not been possible to evaluate the final product usability with summative testing and real field feedback because the product of this study is not yet on the market place when the writing of this thesis is completed.

9.4.7 Improvements to usability engineering lifecycles

The guidelines improve the existing usability engineering lifecycle definitions in many ways. The main difference of proposed guidelines to lifecycle descriptions is the emphasis on product development process instead of usability processes.

The lifecycles proposed by Nielsen (1994), Mayhew (1999) and ISO 13407 are based on linear steps executed in temporal order, i.e. phases, but it has also non-linear iterative activities, such as parallel design, prototyping and iterative design. These lifecycles lack the aspect of coordination, concurrent design activities and practical constraints, such as product complexity, prototype availability, time and resources. In addition, they do not take out-of-box readiness issues, important for all communication oriented consumer products, into consideration.

9.4.8 Some thoughts on education

In the development of complex products, such as information appliances and smart phones, it is reasonable to expect that a usability engineer has some knowledge but not full expertise on current and emerging design and technology areas. Hence, the ability to apply existing usability methods and coordinate the work becomes crucial. The former is not anything new, meanwhile the coordination of usability work is not well covered or studied.

As we have seen, CE, especially in the development of innovative and usable mobile phones, requires knowledge and know-how from human-factors and usability disciplines as well as awareness of project management methods. This should be understood in the training and education of usability engineers.

9.5 Summary

I have now defined and evaluated general guidelines for usability engineering in CE product development. By defining the requirements I provide a foundation for:

- understanding what is needed for efficient and effective usability engineering in complex product development,
- open the guidelines for further validation, development and instantiations.

A practitioner, usability engineer, should ask, how can I operationalise the guidelines? How can they be applied? Each guideline presents an idea or points to a potential problem area. For example guideline “Usability engineering needs proper coordination (planning and follow-up).” proposes coordination activities for usability work, with the assumption that without coordination the usability work may not be efficient or effective. A guideline can be operationalised by studying the current project and organisation and asking, can I improve the quality of my work with coordination of usability work?

Usability engineering is seldom a one-shot activity. More often, it is continuous work from one product and project to another. The connection between projects and use of gained organisational memory becomes important. In conceptual level of project execution, each project consists of three separate phases for a usability engineer: data collection, usability work, and knowledge sharing (Article II). The first phase consists of creating and collecting the initial design and requirements data (input) to produce a product. The second phase is the actual development of the product. During this phase the project focuses on creating and using the data in order to create the specific product. In the last phase all data is used as output in order to be used by other projects, and to build the organisational memory (Figure 9.1).

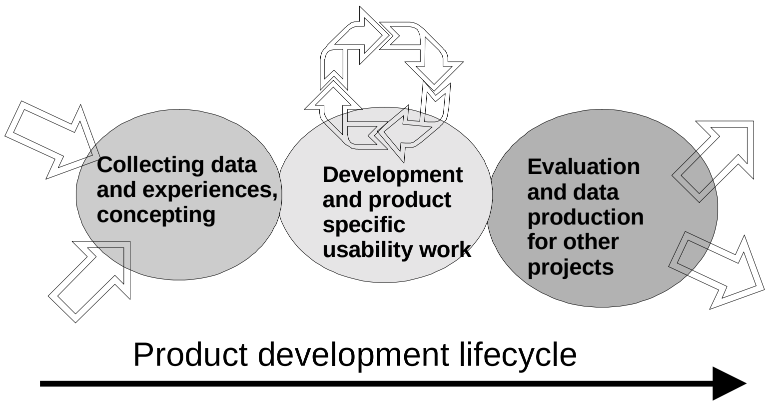

Figure 9.1. Usability work phases

In this chapter I have introduced guidelines for usability engineering that can be used to integrate usability engineering into complex product development. Based on the case study, the guidelines seem to be useful and reasonable. Results are encouraging. I have seen and shown that the efficiency and effectiveness of usability engineering can be improved with the guidelines. I have also shown that the development challenges caused by complexity in development organisation, product structure and user requirements can be overcome with the guidelines.

The proposed guidelines cannot be applied mechanically. They need to be adapted to the specific development organisation and work practices. The guidelines have proven to be useful in the large multi-site development setting with continuous and extensive product development. They may not be as useful for smaller companies and simpler products. The guidelines alone are not enough to perform efficient and effective usability engineering. For each instantiation, they need to be turned to more practical usability engineering methods, practices and strategies. The set of guidelines encourages researchers to consider those systematically in order to improve existing usability engineering methods to meet the complex development domain of smart products and information appliances.

10 Discussion and conclusions

After following and participating three communicator development projects at Nokia Mobile Phones as user interface designer and human-factors engineer, I was given a tempting opportunity to participate as usability engineer in the development of a highly innovative smart phone. The earlier experiences had shown me the challenges in creating usable products in the particular development setting of NMP. The previous experiences and the project at hand, together, led me to search for and develop better usability engineering practices that would work with concurrent product development. The new development project gave me opportunity to be an insider researcher and to perform action-research. During the numerous usability experiments and design activities I found the idea of Ackoff (1974. 8) to be very true: "We fail more often because we solve the wrong problem than because we get the wrong solution to the right problem". In the middle of complex product development it is easy to lose the focus and work on secondary design challenges.

I formulated the research problem as follows: How can usability engineering be effectively and efficiently performed in a CE mobile phone development project? The research problem was further divided to six research questions. The basic flow of research activities is initially described in sub-section 1.2.3. In the very beginning of joining the development project (at M0) I needed to evaluate the success of usability engineering in previous projects and analyse reasons for existing problems. Some of the lessons-learned were reported in Article I. The analysis and discussions with project management and designers guided me in formulating Usability Assessment Meeting and creating the Usability Plan with Usability Risk Analysis. The findings this far led to the creation of Article II. The Usability Plan was executed and improved according to success and failure in real development projects. As a background activity a new usability process for CE was developed and put in practice. When the Usability Plan seemed to be mature enough it was turned to template documentation and handed out to other development projects, which could then start applying it. The success of this approach was then reported in Articles III, IV and V. The overall experiences and lessons-learned are finally reported in Article VI. Because the articles present an industrial experience, I wanted to extend the articles considerably in this thesis in order to provide appropriate linking to existing research and related work in other disciplines, and also to turn out the lessons-learned as guidelines for further development.

10.1 Research results

This study provides a basis for understanding the rich and challenging development arena of mobile information devices by thorough review on product, organisation, process and user. The main research result of this study for other researchers and practitioners, and answering to the original research question, is the identification of new methods, usability engineering guidelines for concurrent and complex product development, which now can be used for understanding what is needed for efficient and effective usability engineering in concurrent product development, and validated, developed and instantiated in other experiments.

In this study I have also identified the need to improve existing usability engineering lifecycle descriptions by separating usability coordination and execution tasks. Coordination tasks enable the development organisation to plan and perform follow-up in order to ensure that the developed product will be usable. Execution tasks, separated from coordination, enable easier resource allocation, for example in the form of subcontracted usability testing. Some useful concepts were also clarified and created:

- *design uncertainty*

- *user interface, external interface, service interface*
- *insufficient and sufficient usability engineering*
- *possible levels of usability engineering*
- *Usability Plan*
- *Usability Assessment Meeting*
- *Usability Risk Management*
- *efficient usability engineering*
- *effective usability engineering*
- *Usability Milestones*
- *early and late design phases and motivation of development organisation*
- *Usability guidelines.*

CE is promoted as a process that takes the customer expectations and requirements into account (for example, Cleetus 1992). Earlier research has not given evidence that users can really be involved in concurrent product design. This study is evidence that users *can* be involved. CE in this case study was performed in a complex organisation with a complex product and complex user requirements. According to the findings in this dissertation I define the role of usability engineering in CE project as follows:

The role of usability engineering, in addition to traditional usability engineering tasks, in a CE project is to:

1. Improve the predictability of timetables by minimizing design uncertainty.
2. Back-up the product development by ameliorating communication between design teams, especially between designers and management.
3. Interpret early user requirements, design sketches and ideas to a form that can be evaluated by user testing and understood by different shareholders.
4. Support the product development in achieving customer satisfaction.

The proposed guidelines were found to be useful for NMP, improving the success of usability engineering. NMP, and Nokia in large, is not a typical example of an organisation where usability engineering is performed due to the large size, mass market oriented product development, multi-site organisation and highly innovative product development. Organisational characteristics, such as maturity and usability capability, have an effect on the effectiveness and efficiency of usability engineering, and on the applicability of guidelines.

New innovations are and will be continuously created and embedded into personal information products. Product complexity and variety will continue to increase. This is a challenge for usability engineering. How can usability be maintained when the products, organisations and user requirements are increasingly complex? When products are increasingly complex, the users are increasingly helpless in problem solving when problems occur!

On one hand, current research on usability engineering is mostly focused on software development. On the other hand, human-factors research is clearly focusing on ergonomics. The same distinction is also seen in the academic training programs. In the development of personal information products the skills of both disciplines need to be applied in product development.

Usability practitioners need to understand and find ways to deal with the increasing product variety and converging products, and also increasing design uncertainty. User needs become even more important for product development, and the understanding and elicitation of user requirements becomes more challenging than before. One potential improvement to current situation is to apply the proposed guidelines to better manage and coordinate usability engineering of complex products.

By reviewing the conference and workshop themes of ACM SIGCHI, IEEE, Interact and BCS (British Computer Society) during the past 24 months it seems obvious that there is increasing interest in developing design methods and practices for personal mobile devices. However, there is not much activity on developing human-centred design processes for time-critical complex or CE projects. Researchers of human-computer interaction are needed to analyse, validate and develop further the usability engineering guidelines and proposed new concepts in order to provide foundation for better human-centred design in complex industrial setting.

10.2 Ideas for future work

What, then, should be the future actions for researchers and practitioners?

From the viewpoint of *researchers*, the findings of this study, especially usability engineering guidelines, should first be validated using other research approaches, due to the small number of case projects and only one industry perspective. Secondly, this study addresses some weaknesses in the current understanding and theory of user-centred design (ISO 13407): the lack of coordination and capability to deal with complex development. It should be studied, whether the ideas of this study can improve the theory of user-centred design.

From the viewpoint of *practitioners*, usability engineers in complex projects, the results of this study suggest that usability engineering approaches described in the literature may not be sufficient for effective and efficient usability engineering, and for making usable products. The practitioners should test the usability engineering guidelines in practice and propose improvements. If they are useful, we will more often see products that are easy to use. Companies developing innovative consumer products, such as Nokia, should continuously strive for embedding user-centred development elements in the early and late development activities.

Many chapters in this study identify and propose ideas for future work. Those ideas are discussed at the end of each chapter (Chapters 4-9). There are still wider research issues not described this far. Large organisations, such as Nokia, have typically several concurrent development projects going on in parallel. In other words, several products are being developed simultaneously. In order to minimise overlapping usability engineering work and to enable better use of the organisation's tacit and explicit memory about user requirements and related issues there is a need to develop mechanisms to support co-operative usability engineering across development projects. Can the principles of CE or usability engineering guidelines be used to improve the efficiency of parallel usability work?

In the projects followed in this study, the usability related knowledge has turned out to be difficult to maintain and share. A typical reason for this phenomenon is that when a project ends, the resources are soon allocated to new tasks. The organisation's memory improves if the knowledge of an ending project is transferred forward or stored. What kind of practices would enable efficient usability knowledge transition when a complex development project ends (the last phase in Figure 9.1)?

Usability of a mobile phone is not only product design. As seen in Chapter 5 it covers also external and service interfaces. The special problem area to be solved by manufacturers, service providers and researchers lies is the transparent integration of wireless services and mobile products. In the ideal world, a user cannot (or need not) make distinction between the personal services (content) and the product, i.e. there is no need to spend time on difficult configuration, reading manuals, finding access numbers or calling help-desks. To what extent we can solve this by product design and implementation with the help of human-centred design?

Usability testing of mobile phones is challenging because of the mobile context and the small size of the product. The special research topics that I have found interesting for mobile phone usability testing are:

- the need for such usability testing methods that ensure the carefully preserved product confidentiality during testing. Mobile phones will have functionality that is location dependent, such as available services. Sometimes, usability testing ought to be done in a real mobile context and real places in order to capture the real-life problems. What is needed for usability testing on the field that should preserve product confidentiality?
- mobile phones and smart devices become smaller. The observation of small devices in the test situation is more difficult than observation of larger systems, for example, desktop applications especially in the mobile context. What kind of observation methods and tools can be used without disturbing the user?
- out-of-box readiness should be good for consumer products. More OBR research focusing on the special system set-up, such as the first connection between the product and supporting PC, of smart device (user-, external-, service-) interfaces is needed.

10.2.1 Cycle of innovations

New design ideas are invented and captured always when users are involved in the early development and product testing. Especially human-centred design activities, such as usability testing, often lead the designers to innovative design solutions and fresh ideas.

Innovations do not (always) come by accident. Sometimes they are logical consequences of earlier actions and development. Innovations have the power to trigger the birth of further innovations. The triggered innovations may be in the same or different expertise areas of development. In the optimal case, triggering can potentially lead to a *cycle (or cycles) of innovations*. In this section I will discuss concept, design, implementation and user innovations, and propose the cycle of innovations as topic for further research.

A product concept can be based on L-, E-, I-, or R- (material, technical, human, social, data, information, knowledge, money) ideas or innovations. The innovations may be originated from several sources, such as new enabling technology, convergence of existing technologies, early identified user needs, insights based on competitor analysis, or findings from market research. These *concept innovations* are often the most important reasons to make the new product.

When the development team converts the product concept to detailed product design, it may produce *design innovations*. These can be logical consequences of solving the design challenges of new concepts. Design innovations can be based on ideas and insights, for example, from human-centred design or user requirements that are identified during product development.

Product implementation solves the practical challenges given in the product design, such as better integrated electronics, new programming algorithms and manufacturing processes. Often, product implementation generates such intellectual innovations that can be protected, i.e. patented, more easily than concept or design innovations. These *implementation innovations* are concrete and often easy to document.

By using a product the users can find innovative solutions for reaching their personal goals. The solutions can be based, for example, on finding new cost-effective ways for communication, such as making charge-free phone calls in order to give a notification to a friend (Kasesniemi 2001). The main idea in these *user innovations* is that the functionality has not been identified or intended by the product designers to be used in the way as they are used.

In the large consumer markets, especially when dealing with technically complex devices, the product maintenance and available customer support is a critical part of the service chain and perceived quality. By studying the product maintenance case stories of users and customer contact points the development organization can identify such product development issues that have major impact on the total cost of future products. These studies may lead the organization to create innovations related to either maintenance or better product and service concepts.

By observing users and identifying user innovations manufacturers have an opportunity to implement functions that support user innovations, or build functionality that better meets the real user needs and supports existing or new work practises. This can lead to new concept innovations, and new cycle of innovations (Figure 10.1).

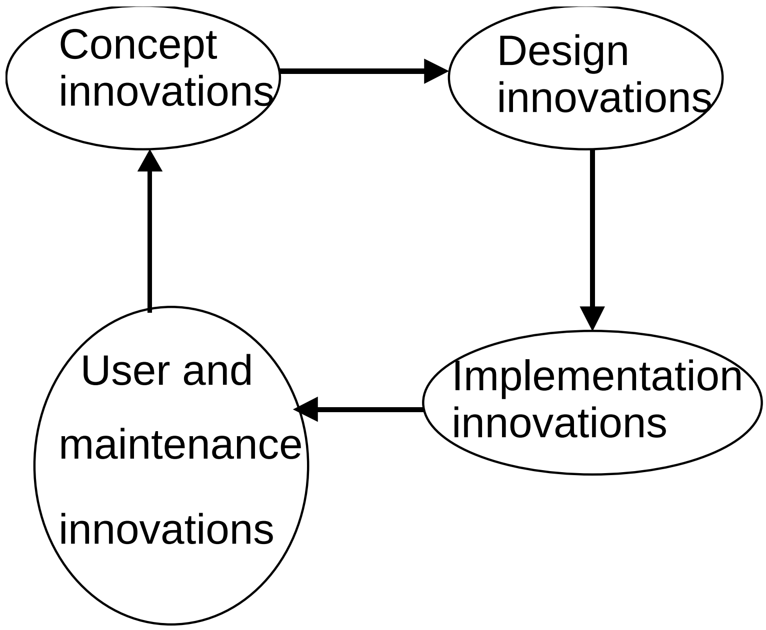

Figure 10.1. Cycle of innovations

Innovations are essential for product development, and an important area of IT research. Any company that has product development and customer research has probably seen triggering of new innovations. Research on the instances of innovation triggering and cycle of innovations might provide useful and valuable information for companies and researchers who need better models and constructs for understanding the birth of new ideas, and to have even more benefit of the human-centred design.

11 Introduction to the Articles

11.1 Article I

Coping with Consistency under Multiple Design Constraints: The Case of the Nokia 9000 WWW Browser

Consistency is a commonly accepted but sometimes problematic design goal. External and internal consistency may conflict, and sometimes the best solution is inconsistent in both respects. We describe user interface design issues and several usability studies for the Nokia 9000 Communicator WWW browser and for WWW pages optimised for the browser. The results show how within the same, restricted design domain, different forms of consistency have to be favoured over others in solving various design problems.

11.2 Article II

Concurrent Usability Engineering

Usability engineering lifecycle models have problems matching with concurrent product development practices. In this article we describe what problems there are between usability engineering lifecycle and concurrent product development process and describe an example how this problem is handled at Nokia Mobile Phones.

Current descriptions for usability engineering lifecycle describe how the work is done during one engineering lifecycle or in a product development project from the very beginning of design to the product launch and to the collection of field feedback. However, in mature development organisations usability engineering is continuous and often parallel work from one product to another and the engineering practice should take this continuity into account. In addition to this, product development is naturally divided to three different phases that set different requirements for the engineering work. These phases are concept work, actual product development and evaluation of the product on the market.

11.3 Article III

Ergonomy and Usability Factors In A Mobile Handset

Usability has traditionally been a user interface issue that deals with interaction between a system and a user. Mobile handsets are devices that extend the concept of a system to be several devices and the user interface more than display and input tool. In this article we define the technical mobile handset user interface components that are involved in user interaction and are thus forming the ergonomy and usability of a handset. The proposed interface structure is based on technical and conceptual mobile handset teardown

11.4 Article IV

The Usability Plan as Part of the Concurrent Engineering (CE) Process: An Empirical Study

How can we direct project usability work to relevant issues when timetables are tight and resources limited? Designing consumer products such as smart phones, it is often impossible to cover the entire product design with available usability resources. To successfully perform Usability Engineering needs the same principles as the project management: planning the work, prioritising the tasks, managing risks and communicating effectively. This article provides a case study from mobile handset industry.

11.5 Article V

What we learned about voice mailbox in Italy? - Experiences about Usability testing at early design phase with low-fidelity prototypes

It is quite challenging to usability test a complex product while the product doesn't yet exist. However, we faced this challenge and travelled to four countries in order to test smart phone functionality with nothing but papers in our hands, literally. The mission was not made easier by the fact that we had to test such user interface concepts that were new both to us and our test users. This article is a short story about a usability test tour. The story takes us to Rome in April. By the side, a more formal description of the test is provided.

A series of usability evaluations was done with paper prototyping technique in order to find out with cultural variables (different countries) how applicable paper prototype testing is for evaluating a new user interface style and new functions.

This study shows that though findings of a single usability test are partially unreliable, they can be assessed by involving cultural and language variables and by repeating the tests. Usability testing is an opportunity for different stakeholders to learn user behaviour and to promote understanding of a new user interface concept.

11.6 Article VI

Usability Engineering Milestones In Complex Product Development - Experiences At Nokia Mobile Phones

How can usability engineering be managed in highly complex innovative product development? When usability engineering is performed in Concurrent Engineering (CE) product development, there are stages where usability engineering needs to be refocused in order to perform successfully in the changing project environment. The focusing points follow the product development milestones but are not identical with those. From usability engineering perspective those points are critical for achieving effectiveness and efficiency in the product development.

References

Abowd G. (2001). Is that a PDA in your pocket, or are you just glad to see me? *SIGCHI Bulletin*, November/December 2001. P. 9.

Ackoff R.L. (1974). *Redesigning the future: A systems approach to societal problems*. John Wiley & Sons, New York

ACM (1992). ACM Special Interest Group on Computer-Human Interaction (SIGCHI) Curriculum Development Group, 1992, section 2.1.

Anderson J., Fleek F., Garrity K. and Drake F. (2001). Integrating usability techniques into software development. *IEEE Software*, Volume 18 Issue: 1, January/February 2001. Pp. 46 -53

Ashby R.W. (1956). *An introduction to cybernetics*. Chapman & Hall, London.

Aulin A. (1982). *The cybernetic laws of social progress*. Pergamon Press, Oxford.

Ayoama M. (1993). Concurrent development process model. *IEEE Software* 1993 Vol 10, No 4. Pp. 46-55.

Back R.J.R. (2002). Software Construction by Stepwise Feature Introduction. In Bert D., Bowen J., Henson M. and Robinsons K. (eds.): *ZB2002: Formal Specification and Development.* Springer Lecture Notes in Computer Science 2272, January 2002. Pp. 162-183.

Bandura A. (1986). *Social Foundations of Thought and Actions: A Social Cognitive Theory*. Prentice Hall. Englewoods Cliffs, NJ.

Beck K. (2001). *Extreme Programmin explained*. Addison Wesley.

Benbasat I., Goldstein D. and Mead M. (1987). The case research strategy in studies of information systems. *MIS Quarterly* 11: 369-386.

Bergman E. (Ed.) (2000). *Information appliances and beyond, interaction design for consumer products*. Morgan Kaufmann. San Francisco.

Bevan N. (1996). Integrating usability into the development lifecycle. Ozok A.F. and Salvendy G. (Eds.). *Advances in Applied Ergonomics. Proc. 1st International Conference on Applied Ergonomics (ICAE'96)*. Pp. 123-128. USA Publishing, Lafayette, IN, USA .

Beyer H. and Holtzblatt K. (1998). *Contextual design: defining customer-centered systems*. Morgan Kaufmann. San Francisco.

Blackburn J.D., Hoedemaker G. and Van Wassenhove L.N (1996). Concurrent software engineering: prospects and pitfalls. *IEEE Transactions on Engineering Management*, Vol. 43, No. 2, May 1996. IEEE. Pp. 179-188.

Blum B.I. (1994). A taxonomy of software development methods. *Comm. ACM* Vol. 37 No 11. Pp. 82-94.

Boland R.J. and Tenkasi R.V. (1995). Perspective making and perspective taking in communities of knowing. *Organisation Science Vol.* 6, No 4. Pp. 350-372.

Brown J.S. and P. Duguid (1991). Organisational learning and communities-of-practice: Toward a unified view of working, learning, and innovation, *Organization Science* Vol. 2, No 1. Pp. 40-57.

Bunge M. (1967). *Scientific research I. The search for system*. Springer-Verlag, Berlin.

Business Week (1990). National Institute of Standards & Technology, Thomas Group Inc., and Institute for Defense Analyses in April 30 1990.

Casaday, G. (1996). Making usability work in the organization. Information technology: Symposium transcription. *Usability Engineering: Industry-Government Collaboration for Systems Effectiveness and Efficiency* (NIST SP 500-237) 219-37

Chao B.P. (1993). Managing user interface design using concurrent engineering. *Proc. Human Factors and Ergonomics Society* 37th Annual Meeting. Pp. 287-290.

Cleetus K.J. (1992). *Definition of concurrent engineering. Technical report*. Concurrent Engineering Research Center, West Virginia University, Morgantown Wv. CERC-TR-RN-92-003.

Cleetus K.J. (1992b). *The concurrent engineering vision*. Concurrent Engineering Research Center, West Virginia University, Morgantown Wv. CERC-TR-RN-92-018.

Cleetus K.J. and Reddy R. (1992). *Concurrent engineering transactions*. Technical report, Concurrent Engineering Research Center, West Virginia University, Morgantown Wv. CERC-TR-TM-92-002.

Cleetus K.J. and Reddy R. (1994). Concurrent engineering transactions. *Proc. CE & CALS Washington '92 Conference & Exposition*, Washington, DC, June 1-4, 1992. Pp. 17-25. Rockford, IL: Society for Computer-Aided Engineering. CERC-TR-TM-92-002.

CockburnA. (2001). *Writing effective use cases.* Addison-Wesley New York.

Cockburn A. (2002). *Project risk reduction patterns.* WWW publication. http://members.aol.com/acockburn/riskcata/riskbook.htm (May 2002).

Coghlan D. (2001). Insider action research projects - Implications for practicing managers. *Management Learning Vol. 32*, No 1, 49-60.

Compeau D.R. and Higgins C.A (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly* Vol. 19, 2. Pp. 189 – 211.

Cooper A. (1999). *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity*. SAMS. Indianapolis, Indiana.

Daly-Jones O., Bevan N. and Thomas C. (1999). *Handbook of user-centred design*. D6.2.1. Version 1.2.

Dataquest. (1999). *Mobile terminals: Worldwide, 1994-2003, market statistics*. MOBC-WW-MS-9916. http://www.dataquest.com.

Diaper D. (1989). Task analysis for knowledge descriptions (TAKD): the method and an example. In Diaper D. (Ed.) *Task analysis for human-computer interaction.* Ellis Horwood. Chichester, UK. Pp. 108-159.

Diaper D. (2001). Task analysis for knowledge descriptions (TAKD): a requiem for a method. *Behaviour & Information Technology* Vol. 20, No 3. Pp. 199-212.

Dix A., Ramduny D., Rodden T. and Davies N. (2000). Places to stay on the move: Software architectures for mobile user interfaces. In *Personal Technologies* Vol. 4 2000. Pp. 171-181. Springer London

Dumash J.S. and Redish J.C. (1993). *A practical guide to usability testing*. Intellect Ltd, Portland, Or.

European Telecommunication Standard ETR 051 (1992). *Human Factors (HF); Usability checklist for telephones. Basic requirements*. European Telecommunications Standards Institute.

European Telecommunication Standard ETR 166 (1995). *Human Factors (HF); Evaluation of telephones for People with special needs; An evaluation method*. European Telecommunications Standards Institute.

Finken S. (2000). Bringing Ethnography Home. In Svensson L., Snis U., SØrensen C., Fågerlind H., Lindroth T., Magnusson M. and Östlund C (Eds.): *Proc. IRIS 23*. Laboratorium for Interaction Technology, University of Trollhättan Uddevalla. Pp. 267 – 277.

Forsythe D. (1999). "It's just a matter of common sense": Ethnography as invisible work In *CSCW Vol. 8*. Pp. 127 – 145.

Fulk J. and DeSanctis G. (1995). Electronic communication and changing organisational forms. *Organization Science* Vol. 6, No 4, 337-349.

Gardner M. (1987). Effects of mood states on consumer information processing. *Research in Consumer Behaviour* Vol. 2. Pp. 113-135.

Gartner. (2001). *Mobile terminals: Western Europe, 1996-2005, market statistics*. Gartner.

Glass R.L. (1995). A structure-based critique of contemporary computing research. *Journal of Systems Software*. Vol. 28. Pp. 3-7.

Gould J.D. and Lewis C. (1985). Designing for usability: Key principles and what designers think. *Comm. ACM*, Vol. 28, No. 3 (1985). Pp 300-311.

Gould J.D., Boies S.J. and Lewis C. (1991). Making usable, useful, productivity – Enhancing computer applications. *Comm. ACM*, Vol. 34, No. 1 (1991). Pp 74-85.

Grossman S. J., Lynch G. and Stempski M. (1992). *Team approach improves user interface for instruments*. *EDN*, June 4, 1992, Pp. 129-134.

Grudin, J. (1989). The case against user interface consistency. *Comm. ACM Vol. 32*, No. 10 (October 1989). Pp. 1164-1173.

Grudin, J. (1992). Consistency, standards, and formal approaches to interface development and evaluation: A note on Wiecha, Bennett, Boies, Gould, and Greene. *ACM Transactions on Information Systems Vol. 10*, No. 1. Pp. 103-111.

Göransson B. (2001). *Usability design: A framework for designing usable interactive systems in practice*. IT Licentiate theses. Dept. of Information Technology, Uppsala University, Sweden.

Hakiel S. (1997). Usability engineering and software engineering: How do they relate? In Smith M.J., Salvendy G. and Koubek R.J (Eds.): *Advances in Human Factors/Ergonomics, 21B Design of Computing Systems: Social and Ergonomic Considerations.* Proc. Seventh Int. Conf. on Human-Computer Interaction, San Fransisco, California, USA. Vol. 2. Elsevier. Pp. 521-524.

Hakiel S. (1997b). Delivering ease of use. *Computing & Control Engineering Journal* April 1997. Pp. 81-87.

Hall R.R (2001). Prototyping for usability of new technology. Gaines B.R (Ed.): *Int. J. Human-Computer Studies Vol. 55.* Academic press. Pp. 485 – 501.

Hamel G. and Prahalad C.K. (1994). *Competing for the future*. Harward Business School Press. Boston, Mass.

Hansen A.L. (1997). Reflections on I/Design: User interface design at a startup. *Proceedings of CHI'97: Human factors in computing systems*. ACM Press. Pp. 487-493.

Hargadon A. and R.I. Sutton (1997). Technology brokering and innovation in a product development firm. *Administrative Science Quarterly* Vol. 42 No 4. Pp. 716-749.

Harrison M.C., Henneman R.L. and Blatt L.A. (1994). Design of a human factors cost-justification tool. In Bias R.G. and Mayhew D.J. (Eds.). *Cost-justifying usability.* Academic Press Inc. Boston. Pp. 203-241.

Hedge A. (1998). Design of hand-operated devices. In Stanton N. (Ed.): *Human Factors In Consumer Products.* Taylor&Francis, London. Pp. 203-222

Heikkinen. M. (1997). *A concurrent engineering process for embedded systems development.* VTT Publications 313. VTT Offsetpaino Espoo Finland.

Helsingin Sanomat (2001). Menestyksen muotoilija. *Helsingin Sanomat, Kuukausiliite* Elokuu 4, 2001. Interview of Frank Nuovo. Pp. 34-37.

Hix D. and Hartson H. (1993). *Developing user interfaces*. John Wiley & Sons, Inc. New York.

Hofmeister C., Nord R. and Soni D. (2001). *Applied software architecture.* Addison-Wesley. Upper Saddle River, NJ.

van. Hofstede A.H.M and Verhoef T.F. (1997). On the feasibility of situational method engineering. *Information Systems* Vol. 22 No 6/7. Elsevier Science Ltd. Pp. 401-422.

Honold P. (1999). "Cross-cultural" or "intercultural" - some findings on international usability testing. In Prabhu (Ed.): *Proc. Internationalisation workshop*. Pp. 107-112.

House J. (2000). *Market mayhem: The smart handheld devices market forecast and analysis, 1999-2004*. International Data Corporation, Doc. #22430. Pp. 1-89.

Häikiö M. (2001). *Globalisaatio – Telekommunikaation maailmanvalloitus 1992 – 2000*. Edita, Helsinki.

Höglund M.(1999). Efficient and effective use of effort metrics. *The European Conference on Software Process Improvement (SPI 99)*, Barcelona, Spain.

IBM (2002). *Out of box experience design guidelines*. http://www-3.ibm.com/ibm/easy/eou_ext.nsf/ publish/ 1534. (January 2002)

Iivari J. (1987). A hierarchical spiral model for the software process: notes on Boehm's spiral model. In *ACM Sigsoft Software Engineering Notes.* Vol. 12 No. 1 Jan 1987. Pp. 35-37.

Isensee S. and Rudd J. (1996*). The art of rapid prototyping*. International Thomson Press. ISBN 1-85032-215-5.

ISO 9126 (1991). *Information technology - software product evaluation - quality characteristics and guidelines for their use*. International Standard.

ISO 9241 (1997). *ISO 9241 – 11: Ergonomic requirements for office work with visual display terminals.* International Standard.

ISO 13407. (1999). *Human-centred design processes for interactive systems.* International standard.

ITU-T Recommendation E.161 (1995). *Series E: Overall network operation, telephone service, service operation and human factors*. Int. Telecommunication Union. Telecommunication Standardization Section of ITU.

Järvinen P. (1980). On structuring problems of job design met. In The Development and Maintenance of Information Systems. *BIT* Vol. 20 No. 1. Pp. 15-24.

Järvinen P. (1999). *On research methods*. Opinpaja Oy. Tampere, Finland.

Järvinen A. and Järvinen P. (2000). *On research methods,* (in Finnish). Opinpajan kirja. Tampere, Finland.

Järvinen P. and Kerola P. (1978). Notes on research in systemeering. *In studies in honour of Olavi Bertel Hellman on the occasion of his fiftieth birthday*. Annales Universitatis Turkuensis A 175, 67-73. Turku, Finland.

Järvinen P. and Kerola P. (1999). New characteristics of virtual phenomena, especially of virtual organisations. In Käkölä T.K. (Ed.): *Proc. 22nd Information Systems Research Seminar in Scandinavia (IRIS 22)*: "Enterprise Architectures for Virtual Organisations". Volume 2. University of Jyväskylä. Jyväskylä. Pp. 139 - 147.

Kaderbhai T. (1998). Overcoming inertia within a large organization. In Trenner L. and Bawa J. (Eds.): *The Politics of Usability*. Springer Verlag London. Pp. 35-48.

Karandikar H.M., Wood R.T. and Byrd J. Jr. (1992). Process and technology readiness and assessment for implementing concurrent engineering. *Proc. Second Annual International Symposium of the Nat'l. Council on Systems Engineering (NCOSE)*. Seattle, WA, July 20-22, 1992.CERC-TR-TM-92-004.

Karat J. and Dayton T. (1995). Practical education for improving software usability. Proc. *CHI '95*. ACM. Pp. 162-169.

Karlsson M. (2001). A holistic approach to usability. In Nygård C.-H., Luopajärvi T., Lusa S. and Leppänen M. (Eds.): *Proc. NES 2001, Promotion of Health through Ergonomic Working and Living Conditions*, 33rd Annual Congress, 2-5 September 2001, Tampere, Finland. Institute of public health, Tampere. Pp. 33-37.

Kasesniemi E-L. and Rautiainen P. (2001). *Kännyssä piilevät sanomat (Embedded messages in mobile phones, In Finnish)*. Tampere University Press. Tampere, Finland.

Keil M. and Carmel E. (1995). Customer-developer links in software development. *Comm. ACM Vol. 38*, No. 5. Pp. 33 – 44.

Keil M., J. Mann and Rai A. (2000). Why software projects escalate: An empirical analysis and test of four theoretical models, *MIS Quarterly Vol 24,No 4.* Pp. 631-664.

Keinonen, T. (1997). Expected usability and product preference. *Proc. DIS'97 Conference Designing Interactive Systems*, 18.-20.8.1997 Amsterdam. Pp. 197 - 203

Keinonen T. (2000). Pieniä tarinoita pienistä puhelimista (Small stories about small phones, in Finnish). In Keinonen T. (Ed.): *Kuinka käytettävyys muotoillaan*. Nokia Oyj, Helsinki. Pp. 207-220.

Keinonen T. (2000b). Yksiulotteinen käytettävyys (One-dimensional usability, in Finnish). In Keinonen T. (Ed.): *Kuinka käytettävyys muotoillaan*. Nokia Oyj, Helsinki. Pp. 92-105.

Keinonen T., Nieminen M., Riihiaho S. and Säde S. (1996). *Designing usable smart products*. Helsinki University of Technology, Dept. Computer Science. Report TKO-C81. Otaniemi.

Kiljander H. (1997). *User interface prototyping of handportable communicating products*. Helsinki University of Technology. Otaniemi, Finland. Pages 122.

Kiljander H.(1999). User interface prototyping methods in designing mobile handsets. *Human-Computer Interaction INTERACT '99*. Pp.118-125.

King N. and Majchrzak A. (1997). Development of multi-disciplinary framework for studying generation of new product concepts. *Innovation in Technology Management - The Key to Global Leadership. PICMET '97*: Portland International Conference on Management and Technology, 477. IEL.

Kirvesoja H. (2001). *Experimental ergonomic evaluation with user trials: EEE product development Procedures*. Dissertation. Oulu University, Dep. of Process and Environmental Engineering, Oulu, Finland.

Klein H.K. and Myers M.D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly* Vol. 23 No1. Pp. 67-94.

Kopomaa T. (2000), *The city in your pocket - birth of the mobile information society*, Gaudeamus, Jyväskylä. ISBN 951-662-802-8.

Korhonen P. (2000). Käytettävyystesteistä liiketoiminnan ytimeen (From usability testing to the core business, in Finnish). In Keinonen T. (Ed.): *Kuinka käytettävyys muotoillaan*. Nokia Oyj, Helsinki. Pp. 181 - 192.

Kotler P. (1997). *Marketing Management: Analysis, Planning, Implementation, and Control*, 9th ed. Prentice Hall. Chapter 9. Englewood Cliffs.

Kruchten P. (2000). *The Rational Unified Process An Introduction*. Addison-Wesley. Addison-Wesley. Upper Saddle River, NJ. Second edition.

Kuluttajavirasto (2001). Uusien viestinpalvelujen riskit ja vastuu. *Kuluttajansuoja 5*, 2001. Kuluttajavirasto, Finland.

Kusiak A. and Park K. (1990). *International Journal of Production Research* Vol. 28 No. 10. Pp. 1883-900.

Langefors B. (1966). *The theoretical analysis of information systems*. Studentslitteratur, Lund.

Langley A. (1999). Strategies for theorizing from process data. *Academy of Management Review 24, No 4*. Pp. 691-710.

Lano K. and Haughton H. (1992). Software maintenance research and applications. In Leponiemi (Ed.), *NordData'92 Precedings*, Tampere, Finland. Pp. 123-143.

Lederer A.L. and Prasad J. (1992). Nine management guidelines for better cost estimating. *Comm. ACM* Vol. 35 No. 2 February. Pp. 51-59.

Leontjev A.N. (1981). The problem of activity in psychology. In Wertsch J.V. (Ed.): *The concept of activity in Soviet Psychology*. Sharpe, Armonk, N.Y. Pp. 37-71.

Lindén Kaj (1999). Nokia peittoaa yliopistot. An interview in *Tekniikka and Talous* 4.3.1999. P. 32.

Lyytinen K. (1987). Different perspectives on information systems: Problems and solutions. *ACM Computing Surveys 19*, No 1. Pp. 5-46.

Madachy R.J. (2001). New processes for rapid software development. In Callaos N., Audestad J.A. and Sanchez M. (Eds.): *Proc. World Multiconference on Systemics, Cybernetics and Informatics, SCI 2001*, Volume XI Information Systems Technology. IIIS. Pp. 23-27.

Maguire M. (2001). Context of use within usability activities. Gaines B.R (Ed.): *Int. J. Human-Computer Studies 55*. Academic press. Pp. 453 – 483.

Maguire M. (2001b). Methods to support human-centred design. Gaines B.R (Ed.): *Int. J. Human-Computer Studies 55*. Academic press. Pp. 587 – 634.

Marakas G.M., Yi M.Y. and Johnson R.D. (1998). The multilevel and multifaceted character of computer self-efficacy: Toward clarification of the construct and an integrative framework for research. *Information Systems Research* Vol. 9 No. 2. Pp. 126-163.

March S.T. and Smith G.F. (1995). Design and natural science research on information technology. *Decision Support Systems* No. 15. Pp. 251-266.

Markeset T. and Kumar U. (2001). R&M and risk-analysis tools in product design, to reduce life-cycle cost and improve attractiveness. *IEEE Proc. Annual Reliability and Maintanability Symposium*. Pp. 116-122.

Matero J. (1999). *Ikääntyneet ja vammaiset matkaviestimien käyttäjinä* (Elderly and disabled as mobile phone users, in Finnish). Diplomityö. Oulun yliopisto, Dept. of Process and Environmental Engineering.

Mayhew D.J. (1999). *The usability engineering lifecycle*. Morgan Kaufmann. San Francisco.

Microsoft (2002). *Microsoft Rings in Pocket PC Phone Edition*. Press release Feb. 19, 2002. http://www.microsoft.com/presspass/press/2002/feb02/02-19phoneeditionpr.asp. (March 2002).

Mikkonen M., Autio T., Väyrynen S., Ikonen V. and Heikkilä M. (2001). User and concept studies as tools in developing mobile communication services for the elderly. In Nygård C.-H., Luopajärvi T., Lusa S. and Leppänen M. (Eds.): *Proc. NES 2001, Promotion of Health through Ergonomic Working and Living Conditions*, 33rd Annual Congress, 2-5 September 2001, Tampere, Finland. Pp. 194-197.

Miller R.B. (1971). *Human ease of use criteria and their tradeoffs*. IBM Technical Report No. TR 00.2185. 12 April. IBM Corporation. Ploughkeepsie, NY.

Mohage M.F. and Wagner A. (2000). Design considerations for information appliances. In Bergman E. (ed): *Information Appliances and Beyond, Interaction Design for Consumer Products*. Morgan Kaufmann. San Francisco. Pp. 27-51.

Molich R., Thomsen A.D., Karyukina B., Schmidt L., Ede M., van Oel W. and Arcuri M. (1999). Comparative evaluation of usability tests. In *CHI 99 Extended Abstracts. ACM SIGCHI*. ACM, New York. Pp. 83-84.

Monk A. (2000). Noddy's guide to consistency. *Interfaces No.45* Winter 2000. British HCI Group. Edinburgh. Pp. 4-7.

Mowshowitz A. (1997). Virtual Organization. *Comm. ACM Vol. 40* No. 9. Pp. 30-37.

Mäenpää P. (2000). Digitaalinen kaupunki. *Neljännesvuosijulkaisu Kvartti 1/2000* (Quarterly, in Finnish). Helsingin kaupungin tietokeskus. Helsinki, Finland.

Nielsen J. (1989). Coordinating user interfaces for consistency. *SIGCHI Bulletin 20*. Pp. 63-65.

Nielsen J. (1993). *Usability engineering*. Academic Press. New York.

Nielsen J. (1996). Improving system usability through parallel design. *Computer, Vol. 29* No. 2, Feb. 1996. Pp. 29.35.

Nielsen J. (2001). *Coordinating User Interfaces for Consistency.* Morgan Kaufmann. San Francisco.

Nieminen M. (1998). Käytettävyys - osa systemaattista tuotekehitystä (Usability – part of systematic product development, in Finnish). *Systeemityö 4/98.* Teknillinen korkeakoulu. Pp. 2-4.

Nieminen M and Parkkinen J.(1998). Usability activities in product development. In Vink P., Koningsveld E.A.P. and Dhondt S. (Eds.): *Human Factors in Organizational Design and Management – VI.* Elsevier Science B.V. Pp. 433-438.

Nohria N. and Berkeley J.D. (1994). The virtual organization: Bureaucracy, technology and the implosion of control. In Heckscher and Donnelon (Eds.): *The post-bureaucratic organization: New perspectives on organizational change*. Sage, Thousand Oaks. Pp. 108-128.

Nokia. (1998). *Nokia's annual report 1998*. Nokia, Helsinki.

Nokia (1999). *Applications will build the mobile information society*. http://www.nokia.com/ telecom99/showarticle/ 0,5534,17,00.html (February 2002)

Nokia. (2000). Insight: Business review 2000. http://www.nokia.com/insight/pdf/rev2000en.pdf. Nokia, Helsinki.

Nokia (2000b), *Nokia in the second quarter of 2000*, http://www.nokia.com/investor/2000/2Q/index.html. Nokia, Helsinki.

Nokia (2001). *Industry leaders join forces to build the wireless village initiative*. Press release. http://press.nokia.com/PR/200104/817796_5.html. Nokia, Helsinki.

Norman D. A. (1988). *The design of everyday thing*s. Basic Books.

Norman D.A. (1998). The invisible computer. MIT Press, London.

Nunamaker J.F, Chen M. and Purdin T.D.M. (1991). Systems development in information systems research. *Journal of Management Information SystemsVol. 7* No 3. Pp. 89-106.

O'Grady P. and Young R.E. (1991). *Journal of Design and Manufacturing Vol. 1*. Pp. 27-34.

Oxnevad K.I. (2001). NPDT - The future of design is now. In Callaos N., Audestad J.A. and Sanchez M. (Eds.): *Proc. World Multiconference on Systemics, Cybernetics and Informatics, SCI 2001.*

Parker J.W. (2001). Training to enhance design team performance: A cure for tunnel vision. In Callaos N., Audestad J.A. and Sanchez M. (Eds.): *Proc. World Multiconference on Systemics, Cybernetics and Informatics, SCI 2001.*

Parush A. (2001). Usability Design and Testing. *Interactions, Volume 8.* No.5. Pp. 13-17. ACM, New York.

Pawar K.S. and Sharifi S. (1994). Product/process design: Integrating differentiated approaches. In Ladame M. (Ed.): *Proc. Second International Conference on Concurrent Engineering and Electronic Design Automation (CEEDA 94)*. Pp. 29-34.

Pennell J.P., Bertrand H.E. and Slusarczuk M.M.G. (1989). Concurrent Engineering: an overview for autotestcon. Autotestcon Proceedings *'89: The System Readiness Technology Conference,* Philadelphia, PA, September 25-28. Pp. 88-99.

Platt A.-B. (1999). The usability risk. *IEEE International Workshop of E-Commerce*. IEEE.

Potts C. (1993). Software engineering research revisited. *IEEE Software*. September 1993. Pp. 19-28.

Prahalad C.K: and Hamel G. (1990). The core competence of the corporation. *Harward Business Review Vol. 68* No 2. Pp. 79-91.

Preece J. *et al. (1994). Human-computer interaction*. Addison-Wesley. Reading.

Premkumar G. and King W.R. (1994). Organisational charasteristics and information systems plannning: An empirical study. *Information Systems Research* Vol. 5 No 2. Pp. 75-109.

Pulkkinen. M. (1997). *The breakthrough of nokia mobile phones*. Acta Universitatis Oecenomicae Helsingiensis A-122. Helsinki School of Economics and Business Administrations. HeSe Print.

Raddle K. and Young S. (2001). Partnering usability with development: How three organiations succeeded. *IEEE Software January/February 2001*. IEEE. Pp. 38-45.

Reuters (2002). Look out, Palm, here comes the "brick." Jan 15, Reuters. Helsinki.

Rogers E.M. (1995), Diffusion of innovations (4th edition), The Free Press, New York.

Roos J.P. and Hoikkala T. (2000). Onko 2000-luku elämänpolitiikan vuosituhat (Is 2000s the century for life politics, in Finnish). In Hoikkala T. and Roos J.P. (Eds.): *2000-luvun elämä: sosiologisia teorioita vuosituhannen vaihteesta*. Helsinki. Gaudeamus.

Roozenburg N.F.M. & Eekels J. (1995) *Product design: fundamentals and methods.* John Wiley & Sons, Chichester, England.

Royce W. (1970). Managing the development of large software systems. IEEE WESCON. Reprinted in *Proc. 9th Int. Conf. On Software Engineering (ICSE)*, Monterey, CA 1987. IEEE Computer Society Press, Washington D.C. Pp. 328-338.

Sacher H. and Loudon G. (2002). Uncovering the new wireless interaction paradigm. *Interactions Vol. 9 Issue 1*. Pp. 17-23. ACM, New York.

Sawyer S. (2001). A market-based perspective on information systems development. *Comm. ACM* Vol. 44 No.11. Pp. 97-102.

Schön D. (1983). *The reflective practitioner, how professionals think in action.* Basic Books, New York.

Shackel B. (1991). Usability - context, framework, definition, design and evaluation. In Shackel B. and Richardson S.J. (Eds.): *Human Factors for Informatics Usability*. Cambridge University Press. Pp. 21-37.

Shepherd A. (1998). HTA as framework for task analysis. *Ergonomics* Vol. 41. Pp. 1537-1552.

Shneiderman B. (1998). *Designing the user interface: Strategies for effective human-computer interaction (Third Edition)*. Addison-Wesley, Reading.

Sinkkonen I. (2001). Designing for humans: The first use of a product. In Pantzar E., Savolainen R. and Tynjälä P. (Eds.): *In search for a human-centred information society*. Tampere University Press, Tampere. Pp. 215-233.

The Strategis Group (2000). *Smarthones and wireless PDAs*. The Strategis Group, Inc. April 2000.

Snyder C. (1997). Using usability testing to manage risks. In *Information Technology Symposium February 1996* Gaithersburg MD USA. Industry-Government Collaboration for Systems Effectiveness and Efficiency. NIST SP 500-237. Pp. 133-147.

Stanton N. (1998). Product design with people in mind. In Stanton N. (Ed.) *Human factors in Consumer Products*. Taylor & Francis, London. Pp. 1-17.

Statistics Finland (2001). *On the road to the Finnish information society - Summary*. Helsinki, Finland.

Steinbock D. (2001). *The Nokia revolution*. Amacom, New York.

Suomen tekstiviestimarkkinat 1998 - 2002 (2001). *Liikenneministeriön julkaisuja 2001/20*. Helsinki, Edita.

Susman G.I. and Evered R.D. (1978). An assessment of the scientific merits of action research. *Administrative Science Quarterly Vol. 23*. Pp. 582-603.

Swanson E.B. (1994). Information systems innovation among organizations. *Management Science Vol. 40*, No 9. Pp. 1069-1092.

Symbian (2000). *The Symbian Story*. http://www.symbian.com/story.html.

Symbian (2001). *Why is a different operating system needed?* http://www.symbian.com/technology/why-diff-os.html. January 2002.

Säde S., Nieminen M. and Riihiaho S. (1997). Testing usability with 3D paper prototypes – Case Halton system. Applied Ergonomics Vol. 29 No. 1. Elsevier Science Ltd., Great Britain. Pp. 67-73.

Talouselämä (2001). Nokia-suosituksemme? Vahva m… *Talouselämä* No.28 2001. Talentum Media Oy.

Thagard P. (1996). *Mind, introduction to cognitive science.* The MIT Press. Cambridge, MA.

Tianfield H. (2001). Advanced life-cycle model for complex product development via stage aligned information-substitutive concurrency and detour. *Int. J. Computer Integrated Manufacturing, 2001, Vol. 14,* No. 3, 281-303.

Time (2000). No title. June 19,2000. Vol. 155, No.24.

Trenner L. and Bawa J. (1998). *The politics of usability. A practical guide to designing usable systems in industry*. Springer, London.

Truex D.P., Baskerville R. and Klein H. (1999). Growing systems in emergent organizations, *Comm. ACM Vol. 42* No 8. Pp. 117-123.

Tuikka T. (1997). Searching requirements for a system to support cooperative concept design in product development. In *Proc. Conf. On Designing interactive systems: processes, practices, methods and techniques* (August 18-20, The Netherlands). Pp. 395-403.

Ulrich K.T. and Eppinger S.D. (1995). *Product Design and Development*. McGraw-Hill, New York.

Ulrich K.T. and Eppinger S.D. (2000). *Product Design and Development*. McGraw-Hill, New York.

Valjus P. (1994). Limittäinen tuotekehitys valtaa alaa. *Tekniikan näköalat 1*, 1994. Pp. 26-27.

Vessey I. and Conger S.A. (1994). Requirements specification: Learning object, process and data methodologies. *Comm. ACM Vol. 37* No 5. Pp. 102-113.

Virzi R.A., Sokolov J.L. and Karis D. (1996). Usability problem identification using both low- and high-fidelity prototypes. In van der Veer G. and Nardi B. (Eds.): *Proc. CHI'96: Human factors in computing systems*. ACM Press, New York. Pp. 236-43.

Väänänen-Vainio-Mattila K. and Ruuska S. (1999). Designing mobile phones and communicators for consumers' needs at Nokia. *Interactions, September + October 1999*. Pp. 23-26. ACM, New York.

Väänänen-Vainio-Mattila K. and Ruuska S. (2000). Designing mobile phones and communicators for consumer's needs at Nokia. In Bergman E. (Ed.): *Information Appliances and Beyond, Interaction Design for Consumer Products*. Morgan Kaufmann. San Francisco. Pp. 169-204.

Wall Street Journal (1999). Nokia's communicator needs work before consumers try to go online. *Wall Street Journal 1999, October 5.*

Walsh J.P. and G.R. Ungson (1991). Organizational memory. *Academy of Management Review* Vol. 16 No. 1. Pp. 57-91.

Weinstein A. (1994). *Market segmentation*. McGraw-Hill, New York.

Wikberg H. and Keinonen T. (2000). Design draiverina off-line wearability (in Finnish). In Keinonen T. (Ed.): *Kuinka käytettävyys muotoillaan*. Nokia Oyj, Helsinki. Pp. 193-206.

Wiklund M. E. (1994). *Usability in practice: how companies develop user-friendly products.* Academic Press.

Wiklund M. E., Dumas J.S. and Hoffman L.R. (1987). Optimizing a portable terminal keyboard for combined one-handed and two-handed use. *Proc. Human Factors Society - 31st Annual Meeting*, Santa Monica, CA.

Wilson G. (2001). Secrets of good phones. *BBC News online*. Thursday, 26 April, 2001. http://news.bbc.co.uk/hi/english/uk/newsid_1296000/1296073.stm. (January 2002).

Winner, R.I., Pennell, J.P., Bertrand, H.E., Slusarzuk, Marko M.G (1988). *The role of concurrent engineering in weapon systems acquisitio*n. Institute of Defense Analyses Report R-338. Dec. 1988.

Winograd T. (1995). From programming environments to environments for designing. *Comm. ACM Vol.* 38 No 6. Pp. 65-74.

Yin R.K. (1991). *Case study research: design and methods*. USA. Sage Publications Inc. 165p. ISBN 0-8039-3470-X.

Zuboff S. (1988). *In the age of the smart machine: The future of work and power*. Basic Books, New York.