

Timo Poranen (ed.)

Software Project Management Summaries 2011



UNIVERSITY OF TAMPERE
SCHOOL OF INFORMATION SCIENCES
REPORTS IN INFORMATION SCIENCES 7

TAMPERE 2012

UNIVERSITY OF TAMPERE
SCHOOL OF INFORMATION SCIENCES
REPORTS IN INFORMATION SCIENCES 7
JANUARY 2012

Timo Poranen (ed.)

Software Project Management Summaries 2011

SCHOOL OF INFORMATION SCIENCES
FIN-33014 UNIVERSITY OF TAMPERE

ISBN 978-951-44-8730-9

ISSN-L 1799-8158

ISSN 1799-8158

Preface

This report contains summaries of project management articles published in international scientific journals and conferences. The summaries were written as a compulsory task for the Theory of Software Project Management –course held fall 2011.

The summaries are written in English or in Finnish. The summaries are not in any specific order; only English language summaries are first. All summaries have three sections: Introduction, Results and Conclusions.

We hope that these summaries help students to familiarize themselves into various aspects of (software) project management.

Timo Poranen

Tampere, January 2012

| | |
|--|----|
| Preface..... | i |
| Firms' involvement in Open Source projects: A trade-off between software structural quality and popularity | 1 |
| A case study approach for developing a project performance evaluation system..... | 3 |
| Understanding stakeholders' perspective of cost estimation in project management | 5 |
| Representing the behaviour of software projects using multi-dimensional timelines..... | 7 |
| Authentic leadership for 21 st century project delivery..... | 9 |
| Impact of organisational climate and demographics on project specific risks in context to Indian software industry | 11 |
| The spiritual identity of projects | 13 |
| An initial investigation of software practitioners' motivation | 15 |
| Is agility out there? Agile practices in game development | 18 |
| An evaluation of the degree of agility in six agile methods and its applicability for method engineering | 20 |
| Evaluating software engineering techniques for developing complex system with multiagent approaches..... | 21 |
| Standardized project management may increase development projects success | 23 |
| Software effort estimation as collective accomplishment: An analysis of estimation practice in a multi-specialist team | 25 |
| Is your project turning into a black hole?..... | 27 |
| Agile monitoring using the line of balance..... | 29 |
| The optimization of success probability for software projects using genetic algorithms | 31 |
| A study of mega project from a perspective of social conflict theory | 35 |
| On the impact of Kanban on software project work: An empirical case study investigation | 37 |
| Multi-criteria performance analysis for decision making in project management | 39 |
| The value of software sizing | 41 |
| An industrial case study of implementing software risk management..... | 43 |
| Strategic management tools in projects case construction project..... | 45 |
| The role of team problem solving competency in information system development projects..... | 47 |

| | |
|---|-----|
| The influence of situational factors in software product management: An empirical study | 49 |
| A systematic review of evaluation of variability management approaches in software product lines..... | 52 |
| Rethinking the implementation of project management: A value adding path map approach..... | 54 |
| Impact of corporate strengths/weaknesses on project management competencies..... | 57 |
| Software product roadmapping in a volatile business environment..... | 60 |
| User and developer common knowledge: Effect on the success of information system development projects..... | 63 |
| Agile and Kanban in coordination..... | 65 |
| Human resources management main role in information technology project management | 68 |
| The agile requirements refinery: Applying SCRUM principles to software product management..... | 71 |
| Software process improvement barriers: A cross-cultural comparison | 73 |
| Software development project success and failure from the supplier's perspective: A systematic literature review..... | 75 |
| Project management by early warnings | 77 |
| The impact of knowledge coordination on virtual team performance over time | 79 |
| ABC-Sprints: Adapting scrum to academic game development courses..... | 82 |
| Organizing self-organizing teams | 84 |
| Project management for the 21st century: Supporting collaborative design through risk analysis | 87 |
| Project management and national culture: A Dutch-French case study | 90 |
| Information technology project management within public sector organizations | 92 |
| An integrated approach to project management through classic CS III and video game development..... | 95 |
| Deliberate ignorance in project risk management..... | 97 |
| Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation..... | 99 |
| Managing software professionals | 102 |

Firms' involvement in Open Source projects: A trade-off between software structural quality and popularity

E. Capra, C. Francalanci, F. Merlo, and C. Rossi-Lamastra, Journal of Systems and Software, volume 84, issue 1, pages 144-161, 2011

Background

Open Source is increasingly seen by companies as an approach to developing software business. Contributing code is not the only way to participate in an Open Source software project. In addition to coding, every significant software project – Open Source or not – contains for example testing, fixing, documentation and support tasks. Community-based Open Source projects, which are the focus of this paper, can also benefit from financial, logistic or marketing support from commercial contributors. In this paper, the effect of commercial participants is assessed from two angles in particular: structural design quality of the software and project popularity.

For volunteering participants, Open Source projects provide a forum to gain personal satisfaction from producing high-quality software and the sense of belonging to a community of talented specialists. The participation of commercial entities is commonly motivated by reaching business objectives, and the individuals working on Open Source projects as a part of their job may be under conditions (such as stress, deadlines and schedules) negatively affecting the outcome of their work.

On the other hand, project popularity is often a goal of a commercial participant. Thus, this paper is based on a hypothesis that a commercial contributor has a positive impact on project popularity and a negative impact on structural design quality.

Results

Empirical verification of the research hypothesis is done by analyzing the bytecode from a large number of Java software projects hosted at sourceforge.net and conducting an online survey. The bytecode analysis will provide a set of standard design quality metrics for each project analyzed. The online survey, consisting of mostly multi-choice questions, is aimed at project administrators of the sourceforge.net projects participating in the study.

The participation of firms in Open Source projects may be based on a multitude of reasons, ranging from creating markets for services to be bundled with the software itself, gaining access to new knowledge via the Open Source communities, utilizing the Open Source community as a talent pool for recruitment and creating interoperability between systems all the way to gaining an image of a good "Open Source citizen". The role of a commercial participant often changes over the lifetime of an Open Source project. For example, sometimes an internal closed project, initially totally controlled by the company, is gradually opened up during its lifetime and eventually completely released

to the community, receiving no contributions from any paid company employees.

In this study, to cover the great variety in types of participation, 3 different models of participation are recognized:

1. Code (writing code, fixing bugs, donating existing codebase etc.)
2. Management (suggesting requirements and functionalities, planning, design and coordination activities)
3. Support (non-development activities, e.g. logistics, marketing etc.)

Project popularity is measured by the sourceforge.net ranking index, which is a composite of project pageview count on the web site, number of unique downloads and a metric of development activities such as code commits. The statistical analysis of research data is performed using Structural Equation Modelling.

Conclusions

Results support both the hypothesis of firm participation increasing the popularity of Open Source projects and the hypothesis of higher firm involvement leading to lower internal software design quality.

Results also seem to indicate that commercial participants are committed to the popularity of the project and make an effort to increase the projects popularity. It is worth noting that when picking the sample data set for analysis, 110 projects to which the firms were not participating from the beginning were excluded from the analysis. This is due to the difficulty of establishing whether firm participation is the cause of project quality and popularity, or project quality and popularity are the cause of firm participation.

The authors suggest that one reason for the apparent lower structural quality of Open Source software in projects with commercial participants is the short-term focus of the business objectives, such as pressure to release software in time, often with the attitude of "it better be wrong than late, it can always be fixed later". However, it is worth noting that over medium or long term the benefits of having high internal structural quality in a piece of software positively contributes to bug-proneness, tailorability, usability, reliability and user satisfaction.

Kimmo Rinteelä

A case study approach for developing a project performance evaluation system

Q. Cao and J.J. Hoffman, International Journal of Project Management, volume 29, issue 2, pages 155-164, 2011

Background

Current project performance evaluation systems still measure cost and schedule as the core to determine the success or failure of a project, and it is still not well used in most organizations. An approach in Honeywell FM&T (Federal Manufacturing & Technologies) has begun to investigate a new method based on the understanding of current situations and how project management performance measurement has been used in the organizations, and degree of attention to the performance measurement. The goal is to design a new project performance evaluation system to allow managers the ability to audit a project and target where improvement can be made.

Results

In spite of the multidimensional nature of the project performance, cost and schedule performance measures still remain as the most widely used methods of project performance evaluation by organizations in the real world (p.156). But essentially, studies have found that inappropriate designs and use of the evaluation method will mislead project members, users, and stockholders, etc. Even with the use of performance measurement systems, abuse and under-evaluation still exist due to the: requirement changes, lack of experience, less attention from management, company culture; this impedes the benefits of using performance measurement.

The approach is therefore started with a questionnaire listing the questions (p.156):

Research Question 1: Does the use of project schedules as the sole project performance measure result with the majority of projects at FM&T as being inefficient?

Research Question 2: Will the development and implementation of a new performance management system provide both tangible and intangible benefits for FM&T?

Research Question 3: Will engaging in cross-project learning provide benefits to FM&T?

A case study was initialized to design an efficient and productive project performance evaluation system. Technical managers and more than 10 engineers from different departments were interviewed. Grounded theory was used to ensure the performance model were clearly understood by organization personnel, could be supported by different methodologies, selected data could reflect core business, and the result would be used practically in the future. Results of the interviews were then analyzed and categorized based on the similar characters of all concepts. It illuminates that Project Duration is the key output that has FM&T engineers' best interests in mind, while project inputs included: effort, project staffing, priority, number of engineers, technical and complexity. To move on to the second stage of the research, CRS model of Data Envelopment Analysis (DEA) was adopted in order to provide best practice comparison

for FM&T. CRS identifies inefficiencies due to diseconomies of scale and benchmarks performance against units that are operating under the most productive scale size (p.159). Input and output data from twenty engineers were studied by using DEA model to identify the most efficient projects. Factors causing projects to be inefficient were also investigated by using sensitivity analysis, the data verifies that five projects are considered efficient and fifteen are inefficient; cause of the inefficiency is by input and output slack, and impacted not only by project team but also management and project nature.

Feedback to the three queries in the survey are negative to the old performance evaluation system, which uses project schedules as a sole performance measurement is considered inefficient; positive feedback to the new performance management system and cross-project learning, they could provide both tangible and intangible benefits to FM&T.

Important learning during developing new project performance evaluation systems historically include, projects that were developed from top to bottom in one direction, causing the lack of communication between different engineering functions, and therefore, has strong need for a systematic approach through all functions. Successful, or as we say, efficient projects that were discovered during the research can be set as role models for future use to guide other projects. Benchmarks would be used internally to select best performers to promote within the company, and externally to increase the company's competitiveness in bidding.

The new project performance evaluation system was being used by both Design and Manufacturing Department's right after the completion of the research. New requests were added by the other two departments so they could later adopt the new evaluation system. After which, by implementing the new system, tangible benefits were recognized, such as: new projects showing higher efficiency, project duration has decreased, less sick days, and becoming very useful for resource allocation, personnel management and budget control. Therefore, the entire feedback was positive and encouraging.

Conclusions

A two-step approach in the research are primarily using case studies to gather the project performance metrics, secondly using different methods to analyze the data and target the weakness. The outcome of the research is the new evaluation system, it has already been used at project and team level in FM&T, and it would be used widely for other functions. The future of the new project performance evaluation system is very optimistic, because this new evaluation system can also benefit other organizations who want to improve their: project performance, optimize their projects, and control budgets, especially under the current global economy downturn.

Jing Liang

Understanding stakeholders' perspective of cost estimation in project management

H. K. Doloj, International Journal of Project Management, volume 29, issue 5, pages 622-636, 2011

Background

While a large number of models have been developed on dealing the cost estimation and managing escalations in projects, there is still a crucial knowledge gap in current cost estimation practices across the board. This research firstly analyzed the underlying factors related to the major players involved in project development environment from a cognitive perspective and then established a conceptual model including all the phases over project life cycle. The resulting concept model will provide hands-on training to relevant professionals for the capacity building on the improved cost estimating processes across the industry.

Results

Soft system methodology (SSM) is a system approach aimed to analyze systems with complex and less clear-cut characteristics. It is found on systems thinking, which explores problems in the context of holistic system, and focuses on viewing the interactions between components of systems instead of investigating the isolated components.

SSM involves seven distinct stages to analyze complex and organizational situations in the real world. The stages of applying SSM include:

1. Investigate the unstructured problem and make a proposal of the problem situation.
2. Express the problem situation in the format of rich pictures, which is a graphic representation of the manner one may think about the system.
3. Integrated the rich pictures from different organizational members together to generate an overall rich picture containing perspectives from different organizational members.
4. Represent the objectives that our system has to achieve by root definitions.
5. Compared the real world expressions which shown in the rich pictures in stage 2 with the conceptual model generated in stage 4. The comparison may lead to re-iterate of previous stages. By trial and error in stage 5, a conscious, coherent and defensible model can be accomplished.
6. Identified and implemented desirable changes and feasible activities.

The factors affecting cost estimation in engineering project can be classified into five categories: political, economic, financial, technical and attitudinal concerns. The underlying attributes associated with these five categories against the key project elements have been summarized over three major phases namely inception, tendering and initiation phases.

At the project inception stage, the business development team has to take the political and legislative factors into account. The economic conditions and the project scope or

duration expectations must be taken into consideration in the early stage cost estimation process. Clear client briefing and relevant industrial knowledge are substantial in affecting quality of cost estimation. Project manager is expected to be responsible for estimating life cycle cost and pre-assessing the project at the project inception stage. The quality of cost estimation depends on the details specified in clients briefing.

In tender stage, cost overruns generally come from the increasing use of provisional sum, offering alternative solutions in the project plan, assessing the EOIs, managing contamination and time contingencies. Quality of tenderer is another factor found to have effects on the accuracy of cost estimation.

Most of the uncertainties in the previous stages are expected to be eliminated over the initiation stage. Latent conditions and variations to builders or sub-contractors are the common cause of cost overruns. Market conditions will affect previous cost estimation. The variations in cost estimation are mainly associated with value management, contract management and management of variations. Project manager must realize the difference between project programs and realistic implementation challenges onsite. Relevant control and tracking mechanisms should be in place for appropriate adjustments in the cost budgeting and project baseline estimates.

Conclusions

This research explored all the underlying factors through applying the soft system methodology for the purpose of establishing a benchmark for managing cost overruns in the project. The results of showed that traditional cost estimating principles are basically inadequate in solving accuracy in cost management of modern projects. In project inception stage, an early interaction with the key stakeholders and establishment of the clear lines of communications for sharing professional and project based knowledge are critical. In project tendering stage, incorporation of the contractor's perspectives based on legislative and environmental constraints in the estimation process is a crucial element. In project initiation stage, contractual variations and contract administration become crucial in terms of resulting cost performance.

Wuping Yao

Representing the behaviour of software projects using multi-dimensional timelines

A. Rainer, Information and Software Technology, volume 51, issue 11, pages 1217-1228, 2010

Background

Software projects and their performance are very important though the change in projects over time is not being investigated regularly. It seems that the software community lacks an awareness of the importance of software project behavior. Investigation techniques are often limited in some way: they focus on quantitative data rather than qualitative data; may not present multiple variables concurrently in time, or do not maintain a consistent, low-level temporal granularity. The study of software projects is extremely challenging in terms of the demand on resources, the scale and complexity of the data to be analyzed. These challenges may help to explain the limited number of studies of software project behavior over time. Multi-dimensional timelines is an important technique to handle the representation of large volumes of data gathered from different projects attributes.

Results

From a detailed study of the paper, it was established that a multi-dimensional timeline (MTD) is intended to visualize different sets of data over a time span. The time spans can be of different types e.g. quantitative time series, schedule of phases, sequence of events, some qualitative data. It was also observed that an MDT requires data that can be represented in time using simple geometrical elements. In a broader perspective, an MDT is structured into one or more sections in the visual display; each section provides a space to represent a dataset. A dataset is positioned in the space in a way that is sensible for that type of data. It was also observed that it is a good design to maintain a consistent notation within an MDT. The use of different notations helps to represent change of time for different types of data and entity in a particular MDT.

The paper generally addresses the question that how can multi-dimensional timelines be used to concisely represent the behavior of a set of project level attributes as they vary over time? Decision makers in the product family recognized that the requirements of the specific hardware for a transactional logging restricted the product's market because major releases of the product typically occur in a rhythmic cycle of approx. 18-24 months. This can be subdivided into minor releases which can be delivered as a software alternative to the hardware based functionality.

The undertaken research work shows that an MDT can accommodate different types of data which include decision events, states, quantitative time series, qualitative data that has been categorized according to some classification system. The research work also shows that the MDT can represent changes related to the above mentioned data types. The MDTs can be used to infer change occurring, re-planning, the frequency of recurring change. By using multiple sections in an MDT, different types of changes can be represented concurrently. The structure of an MDT it can represent events occurring

along a temporal dimension. Another important requirement of an MDT is the information to be represented must be located in time. An MDT can be understood as a geometrical structure consisting basically of points, lines and planes, so beyond the requirement that information should be locatable in time. There are also of course a variety of labels attached to these geometrical structures. In principle, the relatively complex qualitative information can be presented on an MDT by coding the qualitative statement into a set of categories and then representing each of these categories on the MDT. There are some limitations to MDTs as well. Currently they are unable to accommodate atemporal information; the amount of information that can be represented on an MDT is limited by the size of the display e.g. size of the paper. Also, MDTs currently do not support the representation of causal information.

With MDTs, it is possible to represent project level attributes as they change over time provided that they can be located in time and can be represented in terms of simple geometrical structures. Changes in attributes are documented at the point in time at which the change occurs. Each attribute of the project needs to be represented on the MDT in some way.

In the field of information visualisation, Baldonado et al. [68] have derived a set of eight heuristics for the use of multiple views in information visualisations. These heuristics were derived from the analysis of existing systems, the authors' own experiences of designing systems, and from participation in discussion at the CHI'98 workshop on information exploration environments. The heuristics are presented here in Table 3. According to Baldonado et al., the first four heuristics (diversity, complementarity, parsimony, and decomposition) provide guidance on the selection of multiple views. These heuristics would therefore apply to the choice and content of sections in an MDT. The last four heuristics (space/time resource optimization, self-evidence, consistency, and attention management) provide guidance on the actual presentation of the visualisation. These heuristics would therefore apply to the graphical presentation of the content of sections of any MDT to the reader. (p. 10)

Conclusions

The article shows that very little attention has been directed at modeling temporal aspects of software project. The article presents in a very informative way the user of multi-dimensional timelines which use both quantitative and non-quantitative data to represent different project attributes with respect to time. The article uses very illustrative diagrams to show how MDT can represent qualitative data as a temporal event and also schedule information visualization. Having said that, there is still room for improvement in the ways MDTs are developed and used, and further research can be carried out in this regard.

Muhammad Faisal

Authentic leadership for 21st century project delivery

B. Lloyd-Walker and D. Walker, International Journal of Project Management, volume 29, issue 4, pages 383-395, 2011

Background

Leadership style and project management values are starting to update from 20th century to 21st century. This is due to the fact that focus on ROI (return on investment) and iron triangle results (time, cost budget and acceptable quality), albeit important, are not seen as the highest viewpoints in project leadership anymore. Instead, a more general benefits stance with appreciation to value measures are taking the place. In addition, the leading generation is changing from Baby Boomers (persons born during post World-War II baby boom) to generation X (1950 - 1980) and generation Y (1980 – ~mid 1990s), shaping the leadership values in an evolutionary way.

Results

Lloyd-Walker and Walker studied several leadership oriented texts and performed a pilot study by interviewing CEO for of a global construction company, 10 experienced alliance project leaders and three unit managers.

Key factor on shaping current and future project management methods is the ongoing process of changing generations in the workplace. Currently, Baby Boomers held the most influential positions, but in the following 10 years generation X will start to dominate management positions. With this change the values, expectations and ways of working will change as well. Generation X is in an important position to be as an example now, so that Generation Y is prepared to take over when the time comes.

The authors found out that traditional leaders tend to manage by exception, by passively being concerned with fixing mistakes or looking what went wrong, instead of concentrating on what went right. In addition, traditional leaders adopt an uncritical view of leadership styles and themselves. The new, transformational leadership contains what the authors refer as four I's: **I**ndividual consideration, **I**ntellectual stimulation, **I**nspirational motivation and **I**dealized influence (p. 385). The last "**I**" meaning gaining trust, respect and confidence of employees to be a role model is a key factor in defining authentic leadership. The leadership style proposed to overtake the traditional leadership styles.

Authentic leaders demonstrate a passion for their purpose, practice their values and lead by example with heart as well as with their heads. Their values include hope, trust and positive emotions. They are self-aware and conscious of how they are perceived by others. They are open to development and change. And most importantly, are true to themselves. Perhaps surprisingly, charisma is not a necessary trait of an authentic leader. Instead, the leader's ability to build and maintain relations is.

The followers of authentic leaders build a strong bond with the leader. Leaders are part of the team and help the team to grow and develop. All this help the leader and followers to trust each other, making them confident of delivery or a promise being held.

The danger in authentic leadership is the belief that being true to oneself is sufficient. The leader's view of what is ethical, moral and just play a key factor in defining authentic leader, and these views are completely subjective. If the leaders values differ from the followers values problems may arise. The authors present four categories to define leaders by altruism/egoism y-axis traits and congruent and incongruent x-axis traits. Congruent behavior with altruistic traits represent the authentic leadership, and, for example, egoist with incongruent behavior can be defined to be an “opportunistic pseudo-transformational leader”. These leaders may mislead their followers as well as them selves by being moral chameleons and adopting an ethical stance only when it feeds their purposes. This can be problematic in cultures which have adopted a belief that greed is something to strive for.

The results of the interviews authors held back up the studies preceding them. Authors found out that a certain company battled the GFC and survived without casualties with the help of the authentic leadership values. Other interviews reported that there is a need for increased amount of communication and relationship skills. As companies are largely dependent on other companies the co-operation and trust between them is important.

The authors propose what they call a VAT-model, where leadership traits are divided into three categories: values, affective (commitment) and trust. These categories include the traits of an authentic leader presented earlier, and they happen to be the same what was desired from the leader based on the interviews.

Conclusions

Project manager's human skills, ethical-based-values and emotional intelligence are the most valued traits in the 21st century project management, as these authentic leadership factors have been linked to project management success.

In addition, corporate social responsibility and a demand for more ethical and long term view of organization play an important part in shaping new leadership styles. In order for individuals to chance, the organizations have to change as well.

Further studies on this field are required. Going deeper into what additional elements authentic leadership adds to transformational leadership and developing and testing a model which would allow measurement of authentic leadership needs to be studied.

Antero Mäenpää

Impact of organisational climate and demographics on project specific risks in context to Indian software industry

A. Sharma and A. Gupta, International Journal of Project Management, in press, 2011

Background

Software (SW) industry has been acting as back bone for Indian economy for quite some time. Based on current statistics, Indian SW industry seems to enjoy a healthy growth. However, just like any other SW industry, projects in Indian SW industry share similar risks which effect to the schedule, cost and quality of these projects. Several researches have studied SW project risks and suggested methods to evaluate the risk threat mitigate the effect.

The ability of perceiving SW project risks is influenced by organizational climate. According to current article, effect of demographics and organizational climate to the SW project risks has not been studied very well. This article studies how different demographics and organizational climate factors affect to the SW project risks. The study is conducted within Indian SW industry. However the results are useful to understand how these factors effect to SW project risks in any SW industry.

Results

This article has tried to establish a relation between demographics and organizational climate factors and SW project risks. It is done by studying these factors in Indian SW industry. The study identifies the risk factors affecting Indian SW industry. It then identifies the organizational climate and demographic factors present in Indian SW industry. Finally it investigates the relation between two factors.

First of all, a list of 23 SW project risk factors and 17 organizational climate factors was prepared by in-depth interviews with 40 software project managers. Then a survey was conducted using a questionnaire based on the list to identify SW risk factors and organizational climate factors most present in Indian SW industry. The questionnaire was intricately designed to tap the information about i) the personal characteristics of the respondents viz. designation, age and total experience, ii) the risk factors impacting the success of the last executed project and finally iii) the organizational climate factors that were present during the last executed project (p. 3). The results from 300 survey respondents were analyzed.

The major SW risk dimensions and organizational climate dimensions were identified using factor analysis while correlation and regression analysis were conducted to determine the effect of demographics and organizational climate dimensions on the software risk dimensions. Using the factor analysis technique of Principal Component Analysis (PCA), four risk components and four organizational climate components were extracted. A regression analysis was done to validate the following hypothesis: “The

background variables and organizational climate dimensions affect the project specific risk in the software projects” (p. 7). Relationship between organizational climate dimensions and demographics with project specific risk dimensions was established using regression model and results were collected in form of tables as well as in form of mathematical equations (p. 8-10).

Conclusions

The study explores the Indian software industry’s climate and provides comprehensive list of project specific risk dimensions affecting the software projects in India and identifies the relationship between background variables, organization's climate and software risks.

The study was done very systematically using some of the established standard methods. Article covers the study topic very comprehensively and presents the study results very effectively.

Asif Azhar

The spiritual identity of projects

A. Sense and M. Fernando, International Journal of Project Management , volume 29, issue 5, pages 504-513, 2011

Background

Nowadays projects are becoming more and more complex, uncertain and heavily time-limited, so every participant in the project should set a deep connection with others and become inspired by the spirit of the project. In this article, Sense and Fernando mainly talk about the impact of spiritual identity of projects and how to use spiritual identity to improve project performance.

Results

The authors use four moderators to explain the spirituality of project. 1) Organizational culture 2) Person 3) Project work processes 4) Attachment. These four moderators have a great influence in participants' engagement. (pp. 508-509)

The first moderator means the culture around the project. The culture can guide the direction of a project, influence participant' behavior and improve the cohesion of a project team.

The second moderator is about the property of individual, such as personality, IQ or something like this. Every participant plays different role in a project, some maybe leaders and the others maybe followers, they have different experiences in the project.

The third moderator is about the property of the project. It involves the aims and action strategies, resources available, project design and structure.

The fourth moderator is attachment. Attachment means about the outcome, result or service of a project. They all have a link to workplace spirituality.

The result of the authors' research is finding the primary impacts of project workplace spirituality. There are two levels. (pp. 509-510)

The first one is individual level. In individual level, authors find that during the project the sense of well-being will accumulate. They are inspired by the sense of satisfaction and achievement. This sense will improve the communication with other participants and the result of the project. And also at the individual level, the sense of well-being will also improve the participants' mental health during experiencing spirituality in project work.

The second result is at team and organizational levels. At this level, the mainly impact of experiencing workplace spirituality in projects is the development of social capital. Social capital is the degree of how well participants feel themselves to connect to others and can reflect the social ability and social network of a participant. By the achievement in practice, project outcomes and project processes, the participants' social capital will also increase.

Conclusions

This article mainly talks about the constitution and the impacts of the spiritual identity of projects. The research findings are really interesting and useful. If we use these finds into projects, we can truly make a difference.

Huang Bo

An initial investigation of software practitioners' motivation

H. Sharp and T. Hall, in Proceedings of Cooperative and Human aspects on Software Engineering CHASE'09, pages 84-91, 2009

Background

It is well known that lack of motivation can lead to great failings in software projects. In reverse, improving motivation in the project group can have a positive effect on quality. As there are many research papers about motivation in software engineering, none of them really focus on the facts about why software engineers stay in software engineering.

Previous studies have shown, that software engineers are motivated through change, challenge and problem solving. All of these aspects define the nature of software engineering. In that sense it can be concluded that the nature of the job is the key factor in motivating software practitioners.

The meaning of this research was to find out about the factors that do not only motivate software engineers, but also find out why experienced software practitioners stay in software engineering.

Results

The data was collected from twenty-nine software practitioners at SPA2007 conference held in Cambridge, UK. Attendees were self-selected, which may have affected the results as later implied.

The attendees were split to four groups: developers, managers, mentors/coaches and customers. The three-part session was divided to general discussion about motivation, uncovering the elements of and attitudes towards motivation and to plenary session, which consisted of presentation and discussion about the current research in that area.

The attendees were given forms to fill during the session. The form had four sections although the research was focused on the first three. The first section was about common information about recent experiences in software projects. This section was filled at the beginning of the session.

The second section was divided to four questions which were used to find out the aspects of the job to get satisfaction from (question 1), the features of a project that make the attendee to stay in the job (question 2), the factors that keep the attendee in software engineering (question 3), and the factors that make software development worthwhile (question 4).

The third and final section relevant to this paper was about finding out three key elements of motivation from each attendee. Second and third (and of course the non-important fourth part) sections were filled during the second part of the session. The final part of the session was not part of data collection.

The data collected from the forms was categorized to produce results. Examples of how

an answer gets categorized are described in tables five (p. 86), six (p. 86), seven (p. 89) and eight (p. 89).

The results show that the factors that motivate and give most satisfaction are different from the factors that keep software practitioners in software engineering. The most satisfying and motivating factor was *people* in questions one and two, which was shown in answers like “working with high quality people”, “shared values” and “dealing with smart, creative people”.

Other mentioned factors visible in answers to question one (satisfaction) were *quality*, *success*, *learning*, *problem-solving* and *variety*. In question two (staying in the current project/job) the factors were *influence*, *challenge*, *fun*, *quality*, *pay* and *variety*.

Question three showed that the elements of motivation and satisfaction are different from the elements that keep software practitioners in software engineering. The answers to question three (staying in software engineering) were such as “not skilled to do anything else”, “geek coolness”, “I’m good at it” and “experience, capability” which were categorized under *self-image*. *Self-image* got about double the amount of mentions than any other mentioned category like *problem-solving*, *creativity* and *people*.

Answers to question four (factors that make software engineering worthwhile) were hard to categorize: most of the answers were categorized under *creativity*, but the second largest category was *other*. The lack of consisting data could be a result of lack of new ideas, as this was the last question in the second section of the form.

In the last section of the form three elements of motivation was asked from attendees. The results show that *people* is the most important category followed by *learning*, *challenge*, *quality*, *making a difference*, *variety*, *self* and *problem-solving*. It is possible that answers to last section was influenced by the answers to previous sections.

Conclusions

The results of this research both support and differ from the previous researches about the subject. The categories present in both this and previous researches were *challenge*, *benefit* and *problem-solving*. Some categories have become less important like *change*, *science*, *experimentation* and *development-practices*.

The factors of motivation that have gained more importance include *people* and *teamwork*. In addition, *self-image* as a factor does not show in any of the previous researches. This proves that the elements of motivation in software engineering are evolving.

The results of the paper suggest that these changes could be a result from the changed nature of the profession. While technical skills are still important, software engineering has become more people centric. The conclusion is that *people*, *creativity*, *problem-solving* and *self-image* are the most important elements of software practitioners' motivation.

The article itself states that the results are questioned by the fact that the 29 software practitioners that attended the seminar decided to attend themselves. The paper states that the attendees are not “representative sample from the population of software practitioners” (p. 89). I personally think that this is an understatement of the threat to this

research.

The results of the research are completely usable for managers and researchers. The research shows that the elements of motivation have changed and that the results of the previous researches are out-of-date. In addition the results are usable by the software managers. Managers should understand, that software engineers stay in their jobs even though they would not be satisfied by it.

To my experience the willingness to learn and study software engineering during free time is what makes a good software engineer great. And I would think that only the most passionate software engineers are the ones that attend to seminars like SPA2007. Because of this, the attendees were probably people who are passionate about their profession. This research lacks the input from those software practitioners that do software engineering only for the living. Therefore as a manager I would only benefit from this information in case my team would consist of passionate programmers.

Matti Nieminen

Is agility out there? Agile practices in game development

F. Petrillo and M. Pimenta, in Proceedings of SIGDOC 2010, pages 9-15, 2010

Background

With millions of customers and billions in profits, electronic gaming is one of the powerhouses of today's entertainment industry. The creation of a major release is a complex process which involves highly skilled professionals from such diverse fields as software development, design, art, music and human resources, for example. It would seem that completing these increasingly complex projects profitably would require the identification and use of good practices.

Research in software engineering has described many good and proven practices which have been used successfully in traditional software development projects and which should also be usable in game development. However, while there is literature on teaching and working in game companies and a vast collection of works on technical aspects of game development, only a few works on software engineering in game development exist. Furthermore the few existing works advocate the use of waterfall processes such as UP (Unified Process) based on the assertion that it is a standard in the software industry. This in spite of the fact that due to the special aspects of game development process – defining such subjective abstracts as “fun” is impossible, possible game mechanics need to be tested and refined on somewhat working code – the waterfall model cannot be used in its traditional form. For game development the waterfall process is adjusted to the following form: Conception – Specification of the Game – Definition of the story and art style – Technical specifications – Construction – Quality tests – Gameplay tests – Alfa and Beta tests – Final or Gold Version. Each part of the process is independent and generates a product. Defined game rules can be considered as requirements for the process.

As there is very little work on software engineering practices in game development this paper aims to discover which, especially agile method, practices are in use and how widespread their use is. The analysis will be carried out using PMA (Postmortem Analysis) which, unlike in traditional software development, is widely used in game development.

Results

PMA – originally from empirical software engineering – is a technique in which a project is analyzed after it is finished to discern which practices were good and which caused problems. For the analysis, 20 finished and delivered projects were randomly selected from Gamasutra (www.gamasutra.com) postmortem repository. Gamasutra uses Open Letter Template which is divided into three parts: summary, what went right (best practices, solutions, improvements and project management decisions, that contributed to the success of the project) and what went wrong (Both technical and managerial

difficulties, pitfalls and mistakes). Good practices were recognized by using a 3-step system. First the selected postmortems were read and relevant parts were highlighted. Then agile development and gaming industry literature were examined to select the good practices to be used. The practices that were selected are: Qualified team, Belief in the success of the project, Creativity stimulus, Focus on the product, Version control, Using simple tools, Programming good practices, Agile modeling, Defined process, Quality control, Feedback quickly, Good practices of management, and Continuous integration. Lastly the postmortems were reread and whether or not they utilized the selected practices was recorded.

The results of the analysis confirm previous statements about the importance of the team, as “qualified, motivated or cohesive team” was a practice employed by 90% of the projects. Similarly “belief in the project’s success” was found in 90% of the projects. Second most common practices were “stimulus to creativity” and “focus on the project”, both employed in 80% of the projects. Next are “source version control” and “utilization of simple or productive tools” with 65% and 60% utilization respectively. On the other hand, management practices were not widely used in the analyzed projects. Only 45% of the projects had a “defined process”, “good managing practices” were utilized by mere 25% and as little as 10% used “continuous integration”. The rest of the selected practices were as follows: “programming good practices” 55%, “agile modeling” 45%, “quality control” 40% and “feedback quickly” 30%. On average the projects used 7.2 good practices with a standard deviation of 2.6 and as such the results do not agree with some earlier views about the state of the use of good practices in game development.

Conclusions

The results show that many good practices from agile development are already being used in game development. However, it is likely that these practices are not being used deliberately due to the perceived lack of rigor of agile development. In any case full utilization of agile development to game projects should not be difficult to achieve because of the already widespread use of agile practices. If we accept that traditional software industry and game industry face the same major problems then the solutions developed for traditional software should also be of interest for game industry. As such the spreading use of agile methods should be considered by game developers.

Ville Rahikainen

An evaluation of the degree of agility in six agile methods and its applicability for method engineering

A. Qumer and B. Henderson-Sellers, Information and Software Technology, volume 50, issue 4, pages 280-294, 2008

Background

Agile software development methodologies have grown in popularity over the years and agile software development has proven to be a useful approach in managing software development projects. Various agile methods have been developed, but so far no metric for the degree of agility of a method has existed. The paper presents an analytical framework for evaluating agility. This is both useful in deciding which agile approach to apply and in constructing new agile methodologies, or method engineering. Two traditional non-agile methods are also evaluated for comparison.

Results

The paper presents a four dimensional analytical framework, called 4-DAT. Within this framework, four dimensions of a software development method are evaluated: method scope, agility characterization, characterization of agile values and software process characterization.

The method scope dimension evaluates the applicability of the method in different project scopes, environments, and development styles. Agility characterization measures the method's agile characteristics of flexibility, speed, leanness, learning and responsiveness. A mathematical formula is used to convert numeric values to a degree of agility. The third dimension, characterization of agile values, examines the method's emphasis on agile values, such as communication, documentation and collaboration. Finally, the dimension of software process characterization determines the practices that support product engineering and process management.

Six known agile methods and two traditional non-agile methods are then analyzed using the 4-DAT. Out of the evaluated methods, Crystal, XP and Scrum are found to be the most agile, with ASD, FDD and DSDM also having a notable degree of agility. As for the traditional methodologies, while the spiral model exhibits some agile characteristics, the waterfall model is completely devoid of any agility.

Conclusions

The 4-DAT analytical framework is shown to be a reliable metric for measuring agility in software development methodologies. Applying the framework to known agile and non-agile methods yields expected results. 4-DAT can also be applied to new methods and method engineering, making it a valuable tool in software project management.

Janne Repo

Evaluating software engineering techniques for developing complex system with multiagent approaches

E. Garcia, A. Giret, and V. Botti, Information and Software Technology, volume 53, issue 5, pages 494-506, 2011

Background

Multiagent systems (MAS) are used to build the complex systems in which heterogeneous and autonomous entities interact. There are number of methods and framework existed for developing MAS and selection of appropriate one is critical. Evaluation and Comparison of MAS engineering technique is needed in order to make the selection process easier. This paper introduces a framework called Masev (MAS Software engineering Evaluation framework) that analyzes and compares the MAS software engineering techniques. Masev takes into account two approaches, the organization MAS and service-oriented MAS for developing the complex system.

Results

Masev is an online application that analyze and compare different methods, techniques and environments for developing MAS. It provides a quantitative and qualitative comparison based on a simple metric whose parameters can be modified by the users. In this paper first Masev evaluation criteria is described.

Masev evaluation criteria are characterized in two dimensions and the dimensions are 1) Methodology and modeling language 2) development tools. Methodologies are theoretical specifications that provide guidelines to the developers in the development process whereas development tools are used to create the models and the final implementation. Development tools include modeling tool and the implementation tool. (pp. 495-496)

In organizational MAS the first dimension analyzes how the methodology and its model language support the model of organizational MAS. A set of 51 criteria's are defined for the analysis and these are further classified into five dimensions. The structure dimension analyze the structure of the organization, dynamic dimension which represent the evolution of the organization, functional dimension representing organization component goals, normative dimension representing the set of norms that control the organization and environment dimension representing the elements of environment that interact with the organization. Development tool analyzes how the Modeling tool and the implementing tool support the model and implementation of organizational concepts.

In Service-oriented MAS the methodology, a set of 44 criteria's are defined for the analysis. Methodology and modeling language analyzes the way in which the relationship between agents and services are depicted, considered development stages and offers guidelines in this integration. The development tool dimension analyze the communication, definition standards and facilities to model and implement systems (pp. 497-498).

The main functionality of Masev is to analyze and compare the methodologies, modeling and implementation tools. In Masev web site, users can define their own evaluation tools with their own metric parameters. In the first step users need to specify what they want to analyze for example methodology, modeling tool, implementation tool or toolkit (group of methods or tools). Masev displays questionnaires based on the type of the tool selected. The questionnaires are about the support of services and organization offered by the evaluated tool. Users has the choice of viewing, modifying and even updating the evaluated methods and tools of the same type or of different evaluations. They can even filter the comparison data by selecting the experience of the evaluators and also by selecting the default metric values or their own metric values. These comparisons provide a quick overview of the methods and tools analyzed in each category of the evaluation criteria. With all these specifications the user can select the best tool and method to develop the specific application. It also offers a comparative repository of current methods and techniques (p. 500).

Conclusions

Masev simplifies the evaluation and comparison task and it summarizes the most important issues for developing MAS. Using Masev the developers can easily select the most appropriate MAS tool and method for developing the specified application. Also the developers can easily find the similarities and differences between the evaluated tools (pp. 504-505).

Navaneetha Kotha

Standardized project management may increase development projects success

D. Milosevica and P. Patanakulb, International Journal of Project Management, volume 23, issue 3, pages 181-192, 2005

Background

The study investigates effects that a standardized project management (SPM) might have in development projects in high-velocity industries. Standardized project management is defined in the paper as a standardized set of project management practices. Development projects from high-velocity industries include software development and new product development projects. Projects from both types were studied together, because they share enough similarities and often share components (hardware and software) (pp. 181-182).

Results

The used research method was an empirical study that combined qualitative and quantitative methods and the research was split into three steps. In the first step 12 project managers were interviewed and a literature review was made. The purpose of the first step was to develop SPM constructs from a case-study and use its results in following research steps. In the second step a questionnaire was administered in order to test seven hypothesis based on SPM factors found from the case-study research. In the third step multiple follow-up interviews were conducted. The purpose of the third step was to verify and enrich findings from the previous research steps (pp. 181-184).

Seven independent variables on organizational project management level were found in the first research step that have an effect on project success. These seven project management factors were used in formulating seven hypothesis for the second research step: standardized project management process, project organization, information management system, project management tools, project management metrics, project culture and project leadership (pp. 186-187).

As a result from the questionnaire made for 55 project participants in the second research step, three out of the seven factors were found interesting. Standardization of project management process, project management tools and project leadership may contribute to higher project success, still, the impact of these three factors is not very high (p. 187).

Standardization of project management process contributes to improved project success, because the process can be repeated with minimized variation which leads to improved project speed and quality (p. 190).

A standardized set of project management tools was seen as an important factor in the follow-up interviews, because, for example, less-experienced project managers may not have resources or expertise to select their set of tools (p. 189).

Standardizing project leadership leads to having project managers with a skill set preparing them to face problems and they are not left to randomly develop such skills during projects. When given a standardized skill set project managers are more likely to

deliver on cycle time and meet customer satisfaction, quality and cost goals (p. 190).

An interesting detail about the standardization process was found during the first research step, increasing the degree of standardization of the SPM factors beyond a certain company-specific point tends to lower project success rate. Project management practices must have room for variation in development projects, because of the nature of the projects in high-velocity industries (p. 188).

Conclusions

The findings of the study are explained thorough and are backed by solid research data gathered in three research steps. All in all, the study itself and its findings are really interesting.

Kristian Mattila

Software effort estimation as collective accomplishment: An analysis of estimation practice in a multi-specialist team

K. Børte and M. Nerland, Scandinavian Journal of Information Systems, volume 22, number 2, pages 65-98, 2010

Background

The paper examines the process that a team of software professionals follows when estimating the effort of a software project defined as the number of working hours needed to deliver the product, using a judgement-based approach because as the paper states is the most frequently applied method in practice. The focus of the paper is the social interactions that are involved in the estimation tasks. The process of estimating is seen as a collaborative activity in a multi-specialist team and the paper raises the questions of what characterizes the process as collaborative and multi-specialist and which types of work are needed to accomplish the planning task and agree on an estimate. The authors regard software effort estimation as a social practice and highlight the interdependencies between individuals and material resources and between individuals and collectives and furthermore they examine the dynamics of the interactions by which people engage with their social and material worlds in order to see how estimation tasks are understood and accomplished in a team's environment. The notion of distributed expertise is used, which differs from the traditional notion of expertise where expertise is a set of properties, and rather cognition and expertise are seen as constructed in encounters between different individuals and artefacts and as emergent in activities. This paper presents the re-reading of videotaped data lasting 80 minutes in total and after several views the discussion data was sorted depending on its content and excerpts were chosen for in-depth analysis of the interactions. The team had one project manager, two developers and used a requirements document provided by a company customer to estimate the effort needed for the software project. Besides the interactions the use of the requirements document in the effort estimation was also examined. The concepts of orientation, elaboration, clarification and positioning are used to analyse the interactions. Orientation focuses on how participants orient themselves for the task at hand, elaboration refers to the way the team tries to understand the problem, clarification allows the team to reach a common understanding about the problem and positioning refers to the way participants speak and how their changing positions allow different types of expertise to be used. The team provided a very low estimate compared to the actual effort.

Results

The team starts by processing the requirements specification document which is written by professionals so knowledge of terminology is necessary. The team started estimating design and programming aspects of the system by using the requirements specification. Those were perceived as the most familiar sections and thus the easiest to estimate. In the

second phase the activities were covered in depth and estimates were assigned. In the third phase the estimates were added. There is a detailed analysis of how the discussion on the requirements specification shifts from elaboration to clarification and how the users interact and how the orientation and positioning affect the process. One thing to note is that the process is not linear but instead is a sort of interactional loop where estimates are revised. So called “Rules of thumb” which are products of collective ways of knowing are employed by the team and their usage is the product of historically and culturally preserved practices.

Conclusions

As the authors point this work highlights the distributed character of expertise and explores the dynamics of the process of estimating software effort and they note the need for further elaboration of the communication and collaboration between the team. The authors suggest that further research is needed in the use of the requirements specification as an artefact in the estimation process. According to them software effort estimation cannot be regarded as a one-size-fits-all method but rather it is a complex and collective problem-solving activity that needs to be modified based on the situation at hand.

Mirjan Merruko

Is your project turning into a black hole?

M. Keil and M. Mähring, California Management Review, volume 53, number 1, pages 6-31, 2010

Background

IT-projects are prone to delays, budget overruns and even complete failure. This is largely due to intangible and (usually) complex nature of software-intensive systems. Black hole is a term from astrophysics, a vacuum that sucks everything in and nothing comes out. Keil and Mähring use it as an analogy to projects that suck in more and more resources - cost-estimations and schedules prove to be insanely optimistic, and even with extended budgets, some projects yield little or no results. A project does not turn into a black hole all of a sudden. It is a slow process of escalation that can be analyzed to produce list of warning signs so that corrective measures can be taken as early as possible.

Results

In addition to analyzing hundreds of other IT-undertakings (p. 3), Keil and Mähring conducted in-depth case-studies of two projects, both of which were faced with serious problems. In 1990's, EuroBank ordered a new deposit system that was initially estimated to cost around 5 million dollars. In reality, the project cost them 25 million, but did ultimately produce a system that was adopted to use. The other project, database modernization for California Department of Motor Vehicles, sucked in almost 50 million dollars before it was finally terminated. Eventhough the former example was succesfully turned around at last meters, both projects underwent a similar process of drifting towards a catastrophe.

The process of escalation can be divided into three distinct phases: drifting, treating symptoms and finally rationalizing continuation. In the first phase, some fundamental problems emerge, ambiguities or conflicting goals are detected but not resolved. Later on, when drifting turns into treating symptoms, every occurring small problem is seen and dealt with as a separate issue rather than a symptom of something larger, thus the underlying root causes remain unsolved. Over time problems become more and more visible. At this stage however, a considerable amount of resources is most likely already invested in the project, and termination is perceived as highly undesirable option. Stakeholders begin to rationalize continuation. If no reasons to go on with chosen path are found, they are invented. Needless to say, the project is most likely either canceled or completely reassessed soon after. (p. 7)

To prevent escalation process from ever taking place, Keil and Mähring suggest using intelligent project reporting and developing a culture that encourages problem disclosure. By default, people tend to be reluctant to give out negative information, which leads to biased reports and understating the severity of encountered problems. Atmosphere that promotes open discussion about the direction of the project is needed to make sure that conflicts are detected as soon as possible.

In case drifting is already happening, it should be stopped before the momentum grows too great. Freezing the project temporarily or reducing the number of personnel working on the project are better options than trying to solve problems by tossing more money at them. Getting a small, good team of experts to reassess and replan usually gives better results than increasing headcount. After analyzing the situation, clear decision about continuation is needed, "let's keep going for a while longer and see what happens" - philosophy is doomed to fail.

Even projects that are seemingly going well can become black holes because of environmental changes, for example when a competing product makes yours obsolete, or business goals of the sole client shift radically. Should you be unsure about whether to carry on with a project, rationalizing continuation can be avoided for example by increasing the transparency of the project for larger group of stakeholders or having reviews with outside experts. Believing in what you do is not always easily separated from deceiving yourself. Deceiving a greater number of people is harder.

Conclusions

While every project is different, using the framework described above can help managers or executives alike to determine if their project is drifting, and if this is the case, provide guidelines for finding appropriate countermeasures. If the general direction of the project is wrong, increasing the effort does not help. Instead, slowing down or even halting the progress to re-check the chosen course is advised to prevent further escalation.

Jussi Ampuja

Agile monitoring using the line of balance

E. Miranda and P. Bourque, Journal of Systems and Software, volume 83, issue 7, pages 1205-1215, 2010

Background

Monitoring progress is vital in software projects to keep things going according to plans and to handle problems early. Reporting progress to stakeholders needs to be accurate and up to date. In Agile projects the most common methods used are burn down charts, cumulative flow charts and how many user stories are completed and tests passed. There are short-comings to these methods however. Burn down charts and stories completed charts only report how much work is left and do not take into account the work in progress. Cumulative flow charts report work in progress but don't communicate if the project is on schedule.

The authors introduce the line of balance (LOB) method that gives a broader view on the progress of the project. LOB was used already in the 1940's in production monitoring. Its main application areas today are in scheduling activities, construction flow planning and tracking progress of responses to trouble reports. Miranda and Bourque have modified the LOB method to fit Agile environments and therefore control points are derived from the team's actual performance.

Results

The methods used in Agile projects usually tell the team members what they have done and what they will do next. This is not enough information to base managerial decisions upon. The main advantage of LOB is the visibility of progress. It gives insight into what has been achieved and also what was planned to be achieved. LOB indicates the status of the project as it shows the work in progress. A very important piece of information is knowing if there are any hold-ups. This way resources can be allocated at activities identified as needing attention.

Control points are placed at specific stages in a user story's lifecycle. These points are for example "Started", "Designed", "Tested", "Accepted". The time it takes to pass through a control point is estimated to the best of knowledge. For the estimations to be as accurate as possible they are preferably based on data from a version control system. Then a lead-time is calculated to indicate how much longer it takes to proceed from the control point to the finish, for example six days.

Lead times are presented in the Delivery Plan, that comprises of the Release Plan and the Ideal Plan. The Ideal Plan is for keeping productivity in mind but the Release Plan allows changes in backlog and leans on the team's actual performance. In a status chart the Ideal Plan is the baseline to which the Release Plan's backlog is compared to and the management can determine the state of the project in one look. Changes in the project scope that are bound to happen in Agile projects can easily be updated into the plans to reflect the new workload.

The LOB method was applied to Scrum and feature driven development (FDD) projects in this paper to study its usefulness in bringing more awareness to the management. One of the authors implemented the method in a telecommunication company by tracking trouble reports instead of user stories or features. This speaks of the all-round qualities of the method. The authors claim that LOB is suitable for many different types of purposes like tracking the progress of maintenance activities or deployment projects.

The LOB method is very usable in projects with multiple teams which require more long-term planning and control. Each team has its own delivery plan but the project has one plan with common control points. This is more efficient than comparing separate burn down charts. One of the authors also has experience in applying LOB to portfolio management.

Conclusions

The methods commonly used in Agile projects are easy to produce but lack the potential to fully tell what is happening inside the projects. An alternative to these methods is the line of balance proposed by Miranda and Bourque. Yet their approach is minimalistic and focused on performance which is Agile's preference. The benefits of the line of balance method are a greater accuracy in assessing the project status and identifying bottlenecks.

Anna Kamunen

The optimization of success probability for software projects using genetic algorithms

F. Reyes, N. Cerpa, A. Candia-Véjar, and M. Bardeen, Journal of Systems and Software, volume 84, issue 3, pages 775-785, 2011

Background

Based on the previous research it's found that significant part of the software projects can be considered to be failed or present problems in cost, planning or functionality. In fact, cost estimation and prediction of software success are well researched, but a method has not yet been developed which helps project managers determine the largest risk factors and whether it is cost effective to address them. As a respond to this, this research is aiming to develop models that are using optimization technique based on genetic algorithms.

The general idea behind genetic algorithms is as follows: a population of abstract representations (denominated chromosomes) of possible solutions (individuals) to an optimization problem evolves to produce better solutions. The evolution process begins with a population of randomly produced individuals. In each generation all individuals are evaluated by a fitness function, randomly selected and modified (by mutation and crossover) to form a new population that replaces the previous. This process is repeated until satisfying a termination criterion.

Results

The main goal of this research is to develop software project success/risk analysis models which can aid project managers in identifying, analyzing and controlling potential risks during software development. To achieve this there are two objectives set:

OB1: Identify which risk factors are more influential in determining project outcome regardless of prediction model.

OB2: Identify a cost effective investment of project resources to improve the probability of project success.

To meet these objectives, the probability of success relative to cost is used to calculate the efficiency of the probable project outcome. This efficiency is then used as the fitness function in an optimization technique based on genetic algorithms.

In this research existing prediction models were used which were trained on data collected from survey given to Chilean software developers. The survey contained 88 questions organized into the seven risk categories (management, clients and users, requirements, estimation and activity scheduling, project manager, software development process and software development personnel). It also asked if they considered the project they referenced when answering the survey was a success or a failure. The questionnaire was distributed to practitioners from different Chilean organizations and questionnaires describing 140 projects were received.

On earlier research the survey data was used to develop predictive models based on Bayesian classifiers for two target variables using the Necessary Path Condition learning algorithm. The first target variable was the client organization's perception of the success of a project and the second was the development leader's perception. Three different techniques (Bayesian belief network (BBN), Naive Bayesian Classifier (NBC), and Selective Bayesian Classifier (SBC)) were used to construct the models of both these perceptions of project outcome. Therefore, a total of six models will be analyzed and compared.

A genetic algorithm is implemented that takes the current state of the project, defined as the current states of the risk factors used in the predictive model, then algorithmically change the states to see if these changes improve the probability of success given by the predictive model. It will weigh the change in the probability of success with the cost of changing the states of the risk factors to determine if the proposed project state is efficient.

The method used requires three basic components:

1. A predictive model of project success which takes as an input the states of the risk variables of the project and returns its probability of success.
2. A cost model which gives the cost associated with changing the states of any given risk variable.
3. A function which measures the efficiency of the probability of success relative to the cost.

To fill these requirements, existing prediction models are used, a very basic cost model is defined and a measure of efficiency is adapted from the field of education.

A cost is associated with the state of each variable. So the total cost of a solution generated is the sum of the costs of the states. For the purposes of the study, these costs are assigned manually by the user. It was attempted to generate realistic values for the cost, but main focus was on process rather than an actual case study.

The efficiency of a project state can be defined as the relationship of cost to its success probability. The method was originally used to determine the efficiency of an instructional method defined by the relation between mental effort and performance. Here the efficiency is generalized as the relation between a positive and a negative attribute.

Each individual of the genetic algorithms population represents a possible solution to the problem. In this case an individual is considered as an instance for the model. The individual must be represented by a chromosome consisting of a string. In this implementation, each element of the string, called a gene, represents a risk variable of the project. The value assumed by a gene, called the allele, represents the state of the corresponding risk variable.

The fitness function evaluates the chromosomes taking as input an individual and giving a number proportional to its optimality as output. In this case, the efficiency is used as fitness function. The chromosomes that don't satisfy the defined constraints are penalized. In this case, the fitness for these chromosomes is halved to decrease the chance that the algorithm will converge on a local maximum while also avoid their selection in

the evolution process.

To obtain a new generation, parents are selected using the Roulette wheel method, a probabilistic method in which the probability of an individual of being selected for inclusion in the next generation is proportional to its fitness function. The crossover method used is a one-point crossover, where a random position in the chromosome is selected as the crossover point. In this work, the probability of a mutation for a chromosome is 0.1%. When such a mutation occurs, the new state for the gene is selected randomly from the available states.

The first termination criterion is a limit in the number of generations. This research uses a convergence measure based on the evolution of the best and the average fitness of the population. If the difference between the best and the average fitness is less than the defined threshold after a given number of generations, we can assume that the population has converged. In this case we set a threshold of 0.01 over 20 generations.

The genetic algorithm was run using the six prediction models and a unique example case was defined for each model. Each of these cases represents a software project with a low success probability and was based on projects reported in the survey used to build the prediction models. The initial values for states were manually defined so that the obtained success probability was less than 50 percent. For all models the maximum cost is set as a 20 percent increase from the initial cost of the project. All runs were terminated at generation 300 if they had not already converged.

In each execution the algorithm encounters a satisfactory solution to the problem, given by the chromosome with the highest fitness value found in the population. The chromosome information indicates which variables or risk factors must be managed and controlled to improve the project success at an appropriate cost.

Test cases show that this method can identify areas where software could be improved. Method identifies instances where shifts in the project resource allocation, with and without additional cost, result in a greatly improved probability of success. There was also an example in which the method suggests that the current resources are not enough to improve the probability of success. In one of the cases the method shows that neither a shift in the allocation of resources, nor the securing of additional resources would be enough to change the probable failure of the project. During the process of software development, this information would be invaluable in aiding the decision making process of the project manager.

Conclusions

In the study it is shown that by strategic allocation of resources, a project manager can help to ensure the success of a project, often without a large increase in cost. When additional resources are needed, it is helpful to know where they should be allocated and what return on investment they should bring.

It should be noted that the approach taken here is to optimize the probability of success as given by the Bayesian models. The modularity of the method is very advantageous, as models built with different techniques which perhaps offer greater predictive accuracy could easily be substituted for the Bayesian models used in this research. Likewise, since

the cost is not included in the prediction model, but rather a component of the efficiency measure, it is trivial to adopt any software development cost model to be used in this method.

Mikko Arminen

A study of mega project from a perspective of social conflict theory

G. Jia, F. Yang, G. Wang, B. Hong, and R. You, International Journal of Project Management, volume 29, issue 7, pages 817–827, 2011

Background

The object of this article are mega projects which are studied from a perspective of sociology by method of social conflict theory. The aim of the article is to create conceptual model of megaproject. This model includes three propositions 1) mega project is outcome of social conflict, 2) mega project functions as a safety valve in society and 3) mega project has a close relationship with the level of political centralization and economic development.

Results

Research is done on perspective of sociology. The research objective of sociology is human's whole life and research in sociology should be as comprehensive as possible and should focus on the entirety of society. The social conflict theory that is used as a method in this article has different schools. This article uses mainly the theories of Dahrendorf and Coser.

The conflict theory of Dahrenhof believes that society is always a bit in conflict because of the unequal distributions of authority and power within the social organization. The society is divided into two opposite quasi-groups: the ruler and the ruled.

Coser's view of conflict theory points out that conflicts have positive effects by reducing the possibility of the division of the society in two classes and ensuring social continuity..

A mega project can cause a conflict and conflict can cause a mega project. The later case is usually rarely noticed. A conflict can lead to emergence of a mega project that can have many positive functions in solving the conflict. For example China's 4 trillion yuan investment of construction projects proved to be effective to revive the economical problems.

The article makes three propositions about the mega projects: mega project is the product of social conflict, mega projects play a role as safety valve for society and the degree of centralization of state power and economic foundation determines the scale of mega projects.

First proposition states that as long as core values of society are not harmed conflicts can help to improve the society and reveal unreasonable social rules. The improvements are not created by the conflicts themselves. They are created by people when they notice conflicts and they start to try to solve them. In the long run mega project can reduce unemployment and simulate economy.

Second proposition states that conflicts function like the safety valve on a boiler. They went out the hostility and unsatisfaction of people. For example when Taiwan was expelled from United Nations and suffered economic crisis, the ruler Chiang Ching-Kuo took loan and started ten major construction projects that helped to revive economy.

Third proposition states that the scale of the project is related to how much there is a level of centralization in the society. In countries where there is higher level of centralization the people are more willing to sacrifice their individual interests for collective good. However very centralized social structures are not that tolerant for conflicts than the less centralized ones. In more decentralized societies more individual interest have to be taken into account.

However mega project does not always provide solutions for conflicts. If mega project fails it can generate even more problems and in worst case the destruction of the whole society.

The article puts forward three suggestions of how to construct mega projects effectively. First it is important to consider the second important factor. Besides the mega project itself, it is also needed to take into account the needs that it generates, such as venues for Olympic Games.

Second suggestion is to consider the life circle of mega project. It is important not only to invest to the construction of the project but also to the maintenance and operation of the project after it has been completed.

Third suggestion is to use failure science disciplines on mega projects. This means that the cases of failures must be studied to lessen mistakes in the future projects.

Conclusions

The article studied mega projects in perspective of social conflict theory. Such a studies are becoming more and more popular in the world. In the article three propositions were made: 1) mega project is outcome of social conflict, 2) it functions as safety valve in society and 3) it has close relationship with the level of political centralization and economical development.

Based on the analyses the article gave three suggestions for constructing of mega project: 1) the second important factor must be considered before starting the project, 2) making of mega project should be considered from perspective of life circle and 3) the mega project can be improved with the discipline of failure science.

The limitation of the article is the lack of quantitative analysis. It only discusses the topic trough generalization and induction and does not show any quantitative analysis results so the results of the studies are not so precise.

Petri Kotiranta

On the impact of Kanban on software project work: An empirical case study investigation

M. Ikonen, E. Pirinen, F. Fagerholm, P. Kettunen, and P. Abrahamsson, in proceedings of the 16th IEEE International Conference on Engineering of Complex Computer Systems, pages 305-314, 2010

Background

Toyota Production System (TPS) / Lean Management revolutionized the world of manufacturing by focusing on value adding work and introducing concepts like Just-In-Time, One-Piece-Flow and Visual Management. Lean and its principles have been taken in to use on several different fields from accounting to health care.

One key element in achieving a Lean process is making things visual. Kanban is all about making things visual and keeping process flowing. It is a tool used to enhance production flow by creating a pull system. In a pull system the process is based on downstream demand. When applied to software engineering a Kanban pull system is used to make the project progress and problems visual, enhance developers' commitment and make their work self-directing.

The researchers in this paper conducted an empirical study on the perceived impacts of Kanban on software project work. This study was conducted in experimental software R&D setting called Software Factory. The focus of the study was on the team level.

Results

For the purpose of this study they created a framework for assessing the affects from current literature. The framework consisted of nine perspectives that were evaluated by conducting a case study. The framework consisted of the following nine perspectives: documentation, problem solving, visualization, understanding the whole, communication, embracing the method, feedback, approval process and selecting work assignments.

Empirical evaluation of the results was made using video and direct observations. The video observations were captured while the team was working together with the customer including the Kanban board. The direct observations were made by being present in the Factory, but without participating in the work. In addition to these 8 members of the team were interviewed after the project had ended.

Here are the main findings of the case study based on the framework:

- The team kept documentation at a minimum. They wrote the necessary documentation for internal use but didn't waste time in preparing e.g. large specification documents.
- Problems were solved as soon as they occurred and resources were directed to the highest importance problems, because they are the ones creating blockage in the Kanban board. The importance of Kanban as a problem visualizer was noticed.
- In addition to problems the Kanban board visualized the current progress, who

was doing what and how that task was proceeding. The board was also discovered to be a great source of motivation as the tasks in the Verified column started piling up.

- Important finding with seeing the whole was that everyone had a clear picture of what was being done. That meant that developers could start doing any task available. This was achieved by having all the developers present at the customer demo.
- The team communication was enhanced through atmosphere of trust. Problems on the Kanban board initiated discussion among the team as tasks were cut to smaller pieces or blockages were solved. The team working together did all of that.
- The team found Kanban method to be intuitive. However it was noted that proper training was required for the developers to keep the board up to date. It was also noted how important is the altering of the WIP limits and forming the phase columns to fit that particular project.
- Feedback from using Kanban was achieved by chopping tasks into smaller pieces that significantly shortened the feedback cycles. As already noted how Kanban enhanced communication within the team it was concluded that some of that communication contained feedback.
- The best knowledge of the tasks at hand was on the developers themselves so another developer performed the approval process and management was not needed for the approval. Kanban allowed the team to be self-directing by the team being able to define the logic behind the different phase columns and keeping the approval process flowing on the board.
- The only guideline in selecting work assignments was to follow the priority order among tasks. Kanban enhanced developer motivation by enabling them to select the tasks they were interested in or that were related to their previous task. This also enhanced the team's self-directedness.

Conclusions

The conclusion of the case study was there is an indication of considerable benefits in using the Kanban process model. The main benefits include team motivation and the control over the project activities. For managers it is suggested that they use these nine aspects to concretize the abstract environment of the software development process. Great value for the industry lies in Kanban's ability to give a real-time visualization of the project progress.

Erkki Heikkinen

Multi-criteria performance analysis for decision making in project management

G. Marques, D. Gourc, and M. Lauras, International Journal of Project Management, volume 29, issue 8, pages 1057-1069, 2011

Background

It is hard for project managers to analyze the performance measures because each manager has his own performance interests. To control and manage project performance, maybe using the Iron Triangle is not working. Because the paper tried to issue us a new multi-dimensional Project Performance Measurement System to help manager to make decisions, we learn an aggregation tool short named MACBETH.

Results

MACBETH helps manager to handle mega project though project is very complex. The paper tried to explain a new tool for Multi-criteria performance analysis for decision making in project management, it was divided into five parts.

Introduction about MACBETH: There is a complex undertaking for manager to make decision in a project context. Before making decisions, we need to evaluate the quality and complexity about the project. Briefly, complexity will emerge in different types and forms, and they will come from different sources in all walks of intensity because of the industrial sector or object of the project. In a project context, decision making is including these three elements. First, senior management. Secondly, the quantity of information stemming from the project execution. Third, the project manager's own management policy and the value he attributes to information and object.

Scope and research methodology for complex project: How to consider the project if it is complex. Before using MACBETH method, we need to define what is complex project. The notion of complexity is related to the size of project, the number of departments involved, the number and type of stakeholders, the location or the form of contract. Furthermore, it is intended for 'mature' project organizations since it requires the organization to be capable of carrying out performance measures and of ensuring circulation of this information within the management process. (p.1058) After that, project managers have to make decisions. The quality of decisions they made relies on the capacity of the project manager to express abilities. The decision maker should evaluate both the current situation and all kinds of possible evolution of the project.

Problems statements and background on project performance: Defining a performance measurement which suits each decision is very tough. The multi-dimensional face of performance measurement is an effective and efficient project. The paper mentioned Problem Statement (PS). PS1: What are the different performance measurements that allow a complete analysis of the performance of the project? PS2: What are the dimensions of project performance needed to obtain a complete overview of the project situation? PS3: How can project performance measures be aggregated and published to meet different stakeholder's needs? (p. 1059). For PS1, there is project

performance measurement. We need to build a system that can be described a serial of metrics, or criteria about efficiency and effectiveness. For PS2, we should know project performance dimensions. It is vital for project manager to establish a set of criteria. For PS3, it is project performance measures aggregation. It will make systems have right to calculate the performance and multiple criteria.

Towards a multi-criteria project performance measurement system: First, we should have a system overview. The project performance measurement included the tasks followed; the view used to analyze their performance; the type of performance measurement. The model has three dimensions. Then we consider the model as a cube has principle of organization and access to measures, and commensurability meaning fullness. Third, we use KPI to analyze the cube to evaluate the tasks. Forth, using KPI, it helps us to attribute a weight to each criterion, when it comes to MACBETH.

Case study: a project for the manufacture of landing gear doors: This chapter showed us a case study called a project for the manufacture of landing gear doors. It collected project about the need, choice of tasks followed, Metrics design, KPI weighting and performance analysis. MACBETH helps us to underline the weight of decision maker and his opinions and methods.

Conclusions

There is many ways for project management, the more complex project, the more difficult for project manager to make a proper decision, especially for multi-criteria performance analysis for decision making. Because the project decision makers must to put plenty of variables and elements of activities involved the project. From this paper, we learned how to use MACBETH method to make the assumption and analysis of complex project. Through the case of study, we experience how the model they used and MACBETH help decision maker to collect a mass of information and analyze them. Also it told us it would be easier for project manager to see the scope and structure of the project.

Furthermore, we need to remember these four things: the robustness of using MACBETH; the impact of performance indicator interdependencies on the proposition; links between research into project classification and choices for relative KPI weighting; the possibility of cross-aggregated analysis oriented in pairs could be studied. And, we need performance control.

To sum up, if we were decision makers, when it comes to a complex and huge project, analyze it first confidently and try to use this model and MACBETH method to catch the main points of the project. In this way, we can divide a big project into pieces also we won't lose their relations for each other. We do it step by step, the project performance will be better, the project management will be simpler, and the project is easier to plan, control and handle.

Yanzhao Wen

The value of software sizing

***F.G. Wilkie, I.R. McChesney, P. C. Tuxworth, and N.G. Lester,
Information and Software Technology, volume 53, issue 11,
pages 1236-1249, 2011***

Background

The study shows that software sizing has always been a much neglected area in software development industry partly because software developers think that it does not have any significant benefits and also due to the fact that this is the most complex part to calculate in terms of accuracy. Most of the companies use only expert opinions in this area while others use different approaches only to some extent. The purpose of this paper is to highlight the importance of a software sizing and to find out the most appropriate method to do this.

Results

First of all a survey was done on different countries for cost estimation and software sizing and it was found out that these must be done repeatedly so as to achieve maximum benefits from them, also different sizing techniques were discussed.

For experimenting a company was selected and five of its projects were taken into consideration. The sizing technique that was used is called NESMA “Netherlands Software Metrics Association” which adopts a unit of measure called “function point”, calculation is based on the designing of the software. Two simplified versions of NESMA were included so as to find a low cost alternative, Indicative NESMA and Estimated NESMA. (p. 10-11)

For finding the perfect software sizing technique the process was divided into four steps. First software sizing was calculated using NESMA methods, secondly the results were compared to the rankings given to these by the company (internal calculation), thirdly the amount of time and money spend in calculating the software sizing was compared and finally the result was presented to the management of the company with a questionnaire.

Different results were collected in the forms of tables. Company’s internal ranking was taken (p.14), functional counts were calculated (p. 14), efforts were calculated (p. 15), other sizing coefficients were calculated and compared by plotting graph (p.16-20).

After first three steps it was found out that Indicative NESMA is the cheapest method but the percentage of accurate result is not good. Estimated NESMA on the other hand is relatively cheaper to the FULL NESMA and has a good percentage of accurate results.

The questionnaire was presented to the managers of the company and the main conclusion that was drawn by their discussion was that software sizing is indeed very useful tool for planning propose and to estimate the timeline of the project but due to inconsistency in the result they were not fully committed to use this as the only way but were willing to use it as a confirmation tool for their own estimates and in their view Estimated NESMA was better than the other two.

There were also some threats to the validity of the result produced, the first one being that only five projects were selected also there were no sizing data present in the company with which the researched sizing data could be compared but researched data was somehow compared to other appropriate data provided by the company, there was no domain knowledge available and lastly no data models were present without which estimates could not be presented but many of these obstacles were overcome by the research team.

Some other threats were that only one company was selected however the method used can be applied to all small-to-medium size companies and other sizing techniques were not discussed so the result cannot be compared to them.

Conclusions

The presentation and steps of conducting research were very professional and were backed by firm data gathered but lastly as indicated in the article other sizing techniques should have been considered so results could have been compared.

Muhammad Syed

An industrial case study of implementing software risk management

B. Freimut, S. Hartkopf, P. Kaiser, J. Kontio, and W. Kobitzsch, in Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pages 277-287, 2001

Background

As risk management has become better understood more and more corporates have shown an increasing interest towards it. Despite increasing popularity there are still only few reports available on experiences of introducing risk management into an organization.

The paper “An Industrial Case Study of Implementing Software Risk Management” is a case study on the implementation of a specific risk management method. One of its goals is to respond to the lack of empirical study on this subject.

Results

The used risk management method is called Riskit. While analyzing experiences gained from practical implementation special attention was given to specific objectives. These objectives were estimating usefulness and the cost-benefit of the Riskit risk management process.

As the name of the original paper suggest, the empirical investigation is conducted as carefully designed case-study. The reason why Riskit method was chosen for this particular project is its low complexity which makes it ideal for time-constrained projects. One of the critical factors for the successful risk management is the ability to recognize and prioritize stakeholder’s interests. Most risk management approaches struggles to provide accurate estimates. Because of its more theoretical foundations, Riskit successfully avoids these problems.

In order to understand the final results of the case study it is important to take a closer look at the circumstances and the context of the project. This case study involved one of Germany’s largest telecommunication companies. The company was developing a new tool to support service personnel in their task of administrating. Introduction of new technologies added complexity to the project. Again, new tools were expected to increase the project’s productivity.

The Riskit method was introduced to the participants at the workshop. As the result the risk management team was established and it was supported by external vendor Fraunhofer IESE, who had an active role in risk management meetings as well. They provided their expertise by preparing the management techniques to be applied and providing the necessary documents.

The aim of case study was to evaluate the success of the project with respect to research goals. The first goal was to characterize the usefulness and adequacy of the Riskit risk management process. The second goal was to characterize the cost-benefit of the Riskit process from the viewpoint of the department head and the project manager. These two goals were in turn refined into metrics defining the data to be collected to answer the questions and evaluate the research goals. Two types of metrics were identified: quantitative metrics and qualitative metrics. The quantitative metrics included the information that is relatively easy to present in numbers such as time spent on project. The qualitative metrics included aspects like benefit which are difficult to count directly.

During the instantiation of risk management to Tenovis the facilitators made the following observations. It revealed that the project members outside the risk management team were not included in the risk management activities. Another important discovery is associated with the way that new unknown risks were handled. Those risks that were identified later in the project were not systematically included in risk management. Combined with too long intervals between the risk management meetings it made it very difficult for participants to maintain control over the progress of risk management process.

The cost-benefit relationship is one of the major criterions that enterprises are interested in. However, this relationship is hard to express quantitatively. While the cost is easy to measure quantitatively, the benefit is usually hard to quantify. Due to this factor, the ratio between cost and benefit can only be assessed subjectively. To tackle these issues participants were asked to evaluate the results in terms of invested effort and overall impact. They evaluated the impact of risk management on the project in this case study as too low. On the other hand, at management level the existence of more systematic risk management was seen as worth the cost.

Conclusions

Overall, the results of this case study clearly showed that with some improvements the Riskit method can be a very useful addition to the project management. As more and more case studies will be published, the new experiences gained from them will make the Riskit method even more efficient tool.

Elvis Okemou

Strategic management tools in projects case construction project

M. Naaranoja, P. Haapalainen, and H. Lonka, International Journal of Project Management, volume 25, pages 659-665, 2007

Background

Project strategy is a high level plan for achieving the project objectives. A project management strategy involves e.g. teaming and risk management. There are different approaches to project management strategies. For example, Turner defines that strategy should be studied first in larger context and then link lower context strategies to the larger ones. Anttila et al. think that the best approach is to focus on defining and managing the end product (p. 660).

Choosing the right project is important for organizations' business goals. Guidelines for choosing the product are presented. First, the vision of the project gives overall direction. Christenson and Walker say a vision should be understandable, motivational, credible, demanding and challenging (p.660). Second, the sum of services and products, project scope and related scope management includes the processes required to finish the project successfully.

Two different processes are introduced to help with project selection. Innovation process comprises of three phases: front-end phase, offering development and commercialization phase. Front-end phase consists of opportunity identification, opportunity analysis, idea genesis, idea selection and concept and technology development. Strategic process contains measurement, estimation, calculation, comparing and assessing probabilities. Strategy defines the criteria to assess the project success and the risk management plan.

Results

The research method used is a case study of the renovation and partly new construction of a Finnish school. The researchers and community developed the research question "how can we improve the knowledge management in the selected construction project especially in the beginning of the project" (p. 661).

The town was losing inhabitants, so the town vision was to put effort in offering the best services to attract tax payers. The strategy for it was to include schools in art, music and sports. The project was important to the town decision makers.

The project started with idea generation. The proposal was given to town decision makers and they approved it without using formal methods. The decision makers are going to introduce feasibility reports in order to estimate the future actions and not just assess the current situation. Also, no project plans were made before. There was only a short project definition plan directed to decision makers, so the research team made the project plan.

Next, a common vision for the project had to be made up. A workshop was held with every stakeholder (school personnel, parents..) and together they made up the following

vision: "Practical and safe school that supports specializing by multipurpose rooms and that is economical in the whole life cycle as well as good quality" (p. 663).

The town made a definition report which contained for example the need specification and the content and quality of spaces. The definition report gave the project the constraints and a goal. The vision guided the design. Activity cards were given to school personnel to fill to find out about the multiuse purposes. For example, every classroom would also be a teacher's office. Then, a real size mock up was built of the classrooms. This helped the users to understand the design better and test different alternatives.

Conclusions

The strategy of the town didn't guide the choice of project. Naaranoja et al. believe that the town would benefit from analyzing the project proposals formally (p. 664). Also, note that all projects do not benefit from the same methods, such as with the common vision building, because every project is different with different constraints.

Naaranoja et al. tried to plan the whole project from the beginning, but it turned out impossible. It was better to plan it in sequences for every stage - project planning, design and construction. Besides a clear vision, projects need concrete and measurable objectives and strategies for every stage. Researchers conclude that scope management and selection of the project are most important for the project success (p. 664).

Mervi Ollikainen

The role of team problem solving competency in information system development projects

Y. Li , M.-H. Yang, G. Klein, and H.-G. Chen, International Journal of Project Management, volume 29, issue 7, pages 911-922, 2011

Background

Developer always meet a lot of problems in the information system development, not only technical problems but also many other kind of problems, like team management, requirement management problems, how to solve them is the most important concerned in a information system development process. Actually all of them are related, the ability of a team to solve problems that arise before and during the course of the project can help overcome uncertainty of requirement (page 1). This article confirmed that the IS development project managers should consider problem solving expertise when building a team that considers elements that prepare for the development environment creatively as well as coordinate effectively to achieve a quality IS product. (page 1)

Keywords: Information system development projects; Team competency; Problem solving competency; Software quality; Requirements uncertainty

Results

The article was divided in to six parts as follows.

Introduction

Information System development always become failure, one important reason is the development process is full of uncertainty, like the changing requirement, different opinion of different stakeholders, the changing environment and so on. These problems are difficult to be overcome for IS development project.

There comes two major thought from the development literature are explore as to how to address the problem. The thought here is that if risks can be identified then a portfolio of controls can be selected to be certain that the product tracks the specifications and that the specifications still represent the evolving needs of the stakeholders (page 2). These views related to procedures, techniques, climates and cultures.

There still some different perspectives about how to solve the problems, some of them consider the skills and competencies of the project team members and argues that handling changes in a project environment can enhance the achievement of project success (page 2). The article gave us two terms: centrifugal forces and centripetal forces.

Literature review and research model

This parts the article quotes the idea of some other literature to prepare for describing their own research. For the tasks that to solve the problems in IS development process, the IS development team member should know what is a high quality IS, the quality of an information system is related to how well the product satisfies its customers'

requirements. A good IS should satisfied the stakeholders requirement.

Requirements are always uncertain so how to find, understand, and specify user's requirements is the most important for IS development team. A good IS development team should have the ability to deal with these change.

Research methods

The article use Sampling and Constructs to confirm its idea. They authors chose the members of IS development teams if recently completed project as target respondents, the unit of analysis is the IS development project team in which the participant was recently working. And they also find the appropriate research variables for other parts of their study. They use number "0-5" to mean the "strongly disagree" and "strongly agree". And collected the survey to analysis the data.

Data analysis

In this study, minimum sample size checks and a reactive Monte Carlo analysis were performed. The first step in a PLS path analysis is to examine the measurement model and the second step is to assess the structural model. Then they list three tables to show the result of this study.

Discussion and implications

The purpose of their study was to examine the impacts of an IS development team's problem solving competency on the relationship between requirements uncertainty and system outcomes. Based upon the Sheremata's centrifugal and centripetal forces model, they proposed that anticipation reaction mechanisms are two antecedents of developing in the team's problem solving competency. The results suggested that problem solving competency may be implement essential to produce a quality IS product. The differences between their study and previous efforts were that this study particularly examined a meta-competency in the project team and its impact and lessening the negative effects of requirements. They shows the importance of team competency.

Conclusions

This article studied the impacts of an IS development team's problem solving competency on the relationship between requirements uncertainty and system outcomes. They used collecting data and testing the proposed model to validate their point of view. In some ways, team competency is much more important than individual competency.

This study adopts a different lens of competence development suggested by other disciplines to include both preparatory and reactive strategies and argues that the ultimate goal should be to develop the team's problem solving competency so that IS development teams can deal with situation-specific changes and challenges to develop the products as planned. For the IS development project manager, this study points out the importance of team problem solving competency for IS product quality. This study also provides possible actions to practitioners for developing team problem solving competency.

Qin Yameng

The influence of situational factors in software product management: An empirical study

W. Bekkers, I. van de Weerd, S. Brinkkemper, and A. Mahieu, in Proceedings of Second International Workshop on Software Product Management, pages 41-48, 2008

Background

The researches before in the field of software product management (SPM) and software process improvement almost did not consider the situational factors (SFs) of a company. It will allow small and medium sized companies to improve their software product management by taking that context into account. This research indicates what situational factors are and how important they are influencing the selection process of method fragments in two different applications: SPM process maintenance and method fragment choice.

Results

This research uses these research techniques mainly to gather information:

Interviews – They call them explorative semi-structured interviews. These interviews are not like the common interviews, they not only concentrate on what the interviews says, but also the new questions which may be brought up during the interviews.

Document study – Gathering the documentations which were provided by product managers in order to get an overview of SPM processes.

Literature study – The first list of SFs will be based upon a literature study. A second literature study will be performed to determine what SFs describe method fragments best, these method fragments will be taken from literature.

Meta-modeling – The information gathered about the SPM processes will be processed by means of a metamodeling technique called process-data diagrams (PDDs), which has been developed to model the processes and deliverables.

What is situational factor

A situational factor (SF) is any factor relevant for product development and product services.

They also listed out the situational factors which were often considered:

Business unit characteristics, which included development philosophy, size of business unit team and size of development team.

Customer characteristics, which included customer loyalty, customer satisfaction, customer variability, number of customer, number of end-users, type of customers. Market characteristics, which included hosting demands, localization demand, market growth, market size, release frequency, sector, standard dominance, variability of feature requests.

Product characteristics, which included application age, defects per year:total, defects per year:serious, development platform maturity, new requirements rate, number of products, product lifetime, product size, product tolerance, software platform.

Stakeholder involvement, which included company policy, customer involvement, legislation, partner involvement.

Situational factor weights

For each SF, They used a scale which ranged from 1 (low) to 7 (high), then they asked the product managers how much influence it had on the four SPM process areas. The result is given by a table.

From that table, we can see that the SFs which got a high level weight were those: Size business unit team & Size development on portfolio management, product roadmapping and requirements management.

Defects per year: serious & Defects per year: total on release planning and requirements management.

Fragment choice

They selected 26 master students from the master Business Informatics at Utrecht University to perform a research about whether the list of SFs is suited to describe method fragments, and if so, which of SFs are best suited to describe a method fragment.

The research clearly indicates several dominant SFs, the most frequent SFs are: 'Customer involvement' and 'Development philosophy'. Two other factors that are less frequent than the previously mentioned SFs but still rise above the majority of SFs are the 'New requirements rate' and 'Release frequency'.

Conclusions

The list of SF weights presents an overview of the most important SFs. The four domains can be divided into two groups. The first group is Portfolio management & Product roadmapping, and the second is Requirements management & Release planning. The value of Portfolio management almost always lies close to that of Product roadmapping when looking at the same SF for both areas. The same holds for the values of Requirements management, which almost always lie close to those of Release planning when looking at the same SF.

While a difference in influence can clearly be found when comparing the groups which the SFs have been divided into. The group 'Customer characteristics' is the most important group. It has great influence on all four domains, and has the highest average SF influence weights in three of the four process areas. The most important SFs within this group are the SFs that represent information about the state of the customers, these SFs are 'Customer loyalty', 'Customer satisfaction', and 'Customer variability'.

The SFs relating to the market have a lower influence, the SFs 'Sector' and the 'Variability of feature requests'.

'Product characteristics' has significant influence on the SPM processes, but the

SF‘Software platform’ is also remarkable because of its very low level of influence.

‘Stakeholder involvement’ is the final group, and is on par with ‘Product characteristics’ in terms of its influence on the SPM processes. Overall the SFs are important, with the exception of partner involvement in the two lower process areas (‘Requirements management’ & ‘Release planning’).

There is still another thing we should notice that all the experts used exactly the same detailed definitions for the SFs, However, the experts did not always share the same opinion about the influence of the SFs. This is reflected in the relatively high standard deviations across the list of SF weights.

Zhang Zhang

A systematic review of evaluation of variability management approaches in software product lines

L. Chen, M.A. Babar, Information and Software Technology, volume 53, issue 4, pages 344-362, 2011

Background

The variability in various artifacts to be used for building different products in a software product line (SPL) must be defined, represented, exploited, implemented, and evolved throughout the lifecycle of that SPL. This is what variability management in this research.

Variability management is a fundamental activity in SPLE and key feature distinguishes SPLE from other software development approaches or traditional software reuse approaches. It needs to be supported by appropriate approaches, techniques and tools and there has been a great deal of research on the challenges however no effort to systematically analyse how variability management approaches have been evaluated and to systematically collect and synthesize the existing evidence regarding the effects of the reported approaches. And this article, as a systematic literature review of variability management in SPLE comes out which will benefit both researchers and practitioners.

Method: We carried out a systematic literature review of the VM approaches in SPLE reported from 1990s until December 2007.

Results

The researchers were talking with 97 papers according to their inclusion and exclusion criteria. The selected papers appeared in 56 publication venues. They found that only a small number of the reviewed approaches had been evaluated using rigorous scientific methods. A detailed investigation of the reviewed studies employing empirical research methods revealed significant quality deficiencies in various aspects of the used quality assessment criteria. The synthesis of the available evidence showed that all studies, except one, reported only positive effects.

First of all, before reporting the findings about the evaluation aspects of the reviewed VM approaches, they provide an overview of the reviewed papers through demographic data, and VM approaches identified from the reviewed paper. (p. 350)

Then in 3.2, it mainly talks about the evaluation methods used in primary studies. The researcher developed a scheme for classifying the evaluation approaches used in the reviewed studies which can be seen in Table 8, page 350. As to evaluation methods, example application is the most frequently used means while other evaluation approaches used are laboratory experiment with software subjects, field experiment, simulation and so on.

Evaluation performed in an industrial setting. (p. 351)The researchers applied the 11 elements of the criteria on the selected papers as all studies were rated “yes” on the

first criterion. And to make evidence about effects of VM approaches, they exclude the studies using research methods like "example applications" through 2 parts: synthesizing evidence (p. 352-353) and studied objects (p. 353-354).

In addition, the researchers open up a discussion on the findings from this systematic review. It revealed that variability management research in software product lines has produced a diverse set of approaches and a large majority of these approaches await rigorous evaluation. This study also revealed that a large majority of the studies (80.41%) report evaluation of the presented approaches using scientifically less rigorous evaluation approaches. The results also reveal a general lack of replicated studies, as 95.60% of the proposed approaches have been evaluated by only one study. Another important issue that this systematic review has revealed is the lack of industrial evaluation of the reported approaches.

Conclusions

The researchers also paid attention to the potential impact of the excluded papers based on the number of citations reported on Google Scholar when excluding the papers without an evaluation component.

The research reveals a few problems: variability management research in SPL has produced a diverse set of approaches; most studies report evaluation using less rigorous evaluation approaches; lack of replicated studies; lack of industrial evaluation of the reported approaches.

However, the findings of this systematic review may have been affected by certain limitations such as bias in selection of the reviewed papers, inaccuracy in data extraction, inaccuracy in assigning scoring to each study on each element of the quality assessment criteria used for this study.

In general, the researchers conducted a systematic review of evaluation of VM approaches in SPLE and they believe the findings can provide an important contribution to practitioners and researchers and they highlight the areas which need immediate attention by researchers to collaborate with practitioner and it should be of interest to the empirical software engineering community.

Ke Da

Rethinking the implementation of project management: A value adding path map approach

Qian Shi, International Journal of Project Management, volume 29, issue 3, 295–302, 2011

Background

The right implementation of project management (PM) can add great value to an organization. However, some organizations have gained a little value from PM due to the fact that they have not introduced and applied project management correctly. Therefore, we cannot assert that PM is useless just according to the result of its misusing. In this paper, the author had put forward an approach named “Value Adding Path Map” (VAPM) which can be used to help an organization optimize the process of implement project management and exploit the largest value of project management in order to maximize the value of PM. (pp. 295-296)

Results

In this research, the author concentrated on the implementation of PM rather than the quantitative measurement of the value of PM. Qian Shi showed the configurations of soft system and hard system both were extremely important in terms of adding value to an organization. Regarding Soft System (SS) and Hard System (HS), SS and HS are quite different from each other. It is as same as planting. We need to have good seeds as well as fertile soil. Good seeds represent hard system. Fertile soil represents soft system. If we spend too much on one configuration while ignoring the other, the result will be lower return with higher investment. Thus, the investment in soft system and hard system should be synchronous. The terms “hard” and “soft” are not new terms in the project management community. Crawford and Pollack (2004) have revealed that the terms “hard” and “soft” are entering the language of project management and many writers have recognized that identification and response to the difference between hard and soft aspects of projects can influence their success. (pp. 296)

To find the best path of implementing PM based on the coordination of SS and HS in order to create the largest value to an organization with the least investment, the author introduced VAPM where can direct an organization to introduce and implement PM in a better way. The VAPM includes two dimensions: the SS configuration and the HS configuration. The configuration of SS consists of the general environment of the implementation of PM, including general management system and the project management culture. On the other hand, the configuration of HS consists of the PM process, PM training and knowledge management as well as the tools and techniques using in the management of projects. (pp. 297)

The author used largely the result of the research findings of Thomas and Mullaly (2008) about the main configuration of SS and main configuration of HS (table below). The result showed that only when the configuration of HS and SS both take effect, PM can bring value to an organization. Based on the result, then the author simplified the

calculation process and made VAPM approach simple for practical application by introducing the calculation to describe the value of the implementation of PM by chart (pp. 297):

$$\text{Value } (P_{ij}) = \text{Value } (P_{kk}) ; \text{ where } k = \min (i; j)$$

Following are the main configurations of Soft System and Hard System (pp. 297-298):

Main configuration of Soft System

S1. The general management system

- S1.1 The project management system is aligned with the business strategy and the structure of general management system is highly project oriented;
- S1.2 The general management system includes a basic foundation for the implementation of project management;
- S1.3 The general management system is very well coordinated with the project management system including the involvement of support functions in the organization;
- S1.4 The strategic planning of the organization and project identification are tightly coupled;
- S1.5 The PMO office has been set up;
- S1.6 Project manager has a clear goal and authority in the process of managing projects.

S2. The project management culture

- S2.1 The organization has highly recognized that project management contributes to the success of the organization;
- S2.2 The project management professionals have a clear career path in the organization;
- S2.3 The project managers are highly respected;
- S2.4 The PMO is regarded as an important department of the organization;
- S2.5 The resources allocation system is highly project oriented.

Main configuration of Hard System

H1. Project management process

- H1.1 The organization has a clear definition of regular portfolio management process;
- H1.2 The organization has a clear definition of regular program management process;
- H1.3 The organization has a clear definition of regular project management process including integration management, scope management, cost management, time management, quality management, procurement management, communication management as well as risk management across the initializing, planning, executing and terminating stages of a project.

H2. Project management training and knowledge management

- H2.1 The organization has a training budget for project management which is aligned with the personal development plans of project management professionals in the organization;
- H2.2 The organization has a regular project management training system for top managers;
- H2.3 The organization has a regular training system for project managers;
- H2.4 The organization has a regular training system for project team members;

- H2.5 The organization has a sound knowledge management system of project management including the learning system to improve the effectiveness of the project management professionals which goes beyond training.

H3. Project management methodologies, techniques and tools

- H3.1 The organization has used cost management software;
- H3.2 The organization has used schedule management software;
- H3.3 The organization has used quality management software;
- H3.4 The organization has used risk management software;
- H3.5 The organization has used other project management software with the function of project resource allocation, communication management etc.;
- H3.6 The organization has used systematic project management methodologies and techniques including procedures and templates in the implementation of project management.

By means of the VAPM approach, an organization can diagnose the effectiveness and efficiency of implementing PM and develop and improvement plan correspondingly via comparing their current PM implementation position with the best path of PM implementation. Regarding the list of main configuration of SS and HS, the author also recommended 5 steps to implement PM for companies which are ready to apply PM into their management (pp. 298 - 301):

1. STEP1: S1.1+S1.2+S1.3+S1.4+S2.1+S2.5+H2.1+H2.2
2. STEP2: S1.5+H1.1+H1.2+H1.3
3. STEP3: S2.3+S2.4+H2.3
4. STEP4: S1.6+H2.4
5. STEP5: S2.2+H2.5+H3.1+H3.2+H3.3+H3.4+H3.5+H3.6

Conclusions

The implementation of project management in an organization will greatly affect the value that they may gain from PM where the configurations of SS and HS play critical roles. This is also the foundation for the author to develop VAPM. The VAPM approach just concludes several main factors and makes it simple for practical application. However, there are still some problem with VAPM: the implementation of PM and the value it adds to an organization is affected not only by internal variables in an organization but also by many external variable outside, the value of PM cannot be easily quantified, need detail research to apply to different industrial sectors. (pp. 301)

Trong Nghia Nguyen

Impact of corporate strengths/weaknesses on project management competencies

Z. Isik, D. Arditi, I. Dikmen, and M.T. Birgonul, International Journal of Project Management, volume 27, issue 6, pages 629-637, 2009

Background

The article presents the hypothesis that company's strengths and weaknesses have influence on project management competencies. It is presented because companies in the project oriented industry are succeeding when projects are successfully managed. Therefore it is necessary to define what are meant by strengths and weaknesses and after that it is possible to investigate if those have influence on project management competencies. Used data in this research was collected from 73 different company in Turkish construction industry.

Results

Data collecting methods were typical questionnaires via e-mail and interviews with company executives. Companies strengths and weaknesses were categorized in three different sections.

The company's resources and capabilities is the first section which is defined by company's tangible and intangible assets. Those include financial resources, technical competencies, leadership characteristics, experience, image in the industry, research and development capabilities and innovation tendencies.

Second section is strategic decisions which means the company's capability to implement chosen strategies in real situations. This study represents strategies which are common to project based organizations. Represented strategies are: differentiation strategies, market/project/client/partner selection strategies, project management strategies, investment strategies and organizational management strategies.

Third section in companies strengths and weaknesses is the strength of its relationships with other involved parties. These parties are clients, labor unions and the government. Having these relations in a good shape helps companies to achieve mutual satisfaction of all the parties involved.

Project management competencies were defined according to earlier studies (p. 630) and the suggestions of construction professionals contacted in a pilot study. Chosen competence areas were schedule management, cost management, quality management, human resources management, risk management, supply chain management, claims management, knowledge management and health and safety management.

Validity of collected data was analyzed by professionals in industry and academics. Also several data analyzing methods were used to make sure that all the chosen sections in

company strengths and weaknesses as well as in project management competencies were chosen correctly. Conclusion was that company's strengths and weaknesses have significant impact on project management competencies as presented in the Figure 1. Strong arrow in the middle indicates the noticed influence between strengths and weaknesses and project management competencies. Lighter arrows presents how the sections explained earlier are structured from mentioned subcomponents.

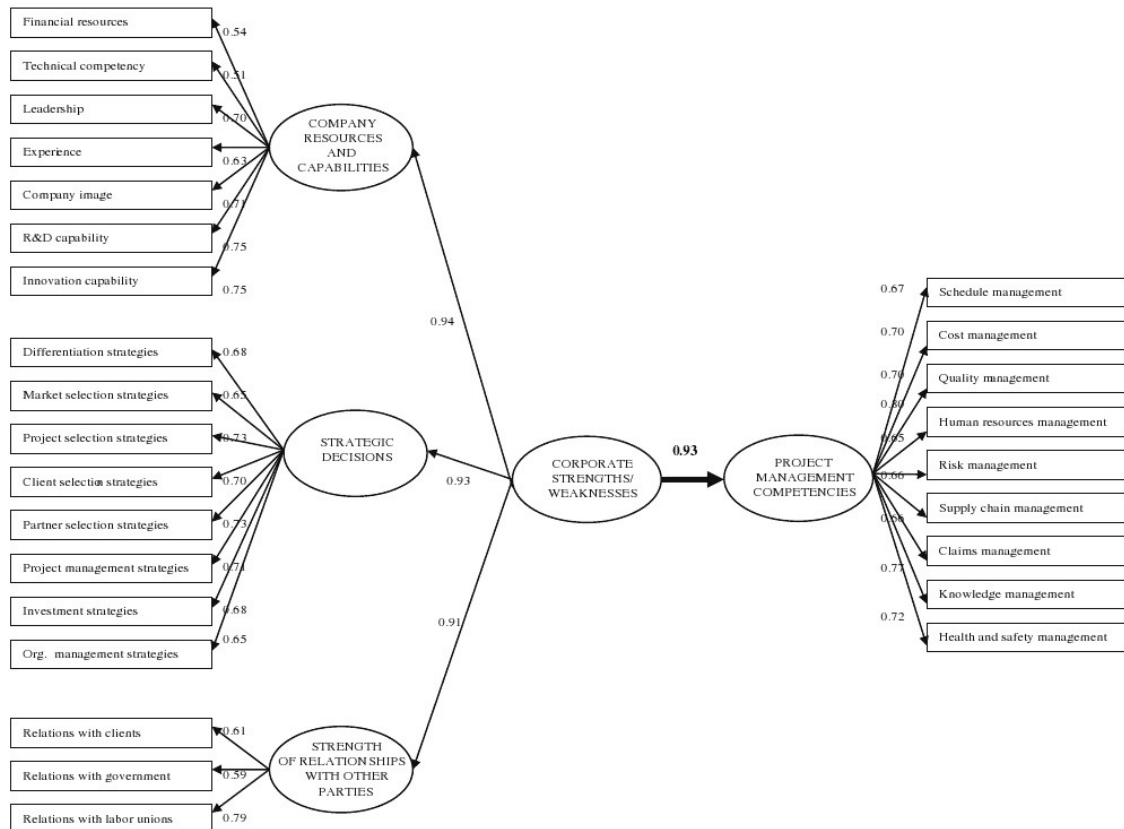


Figure 1 (p. 633).

The accurate structure of “company resources and capabilities” depends on the size of the company and the competitive environment in which the company operates. Some companies may have high research and development capability and gain advantage that way. Other companies may have strong leadership skills and manage to proceed successfully projects without true innovation advantage compared to competitors.

Decision making among the whole company has a significant influence on successfully project management. Company can have capability to choose wisely clients/partners/markets and gain that way better performance on project management. Also financial management and human resource strategies provide great possibilities to successful project management.

Relationships with other parties can be counted as a strength of a company which affects on project management competencies. If the company has negotiated good relationships

with labor unions and government agencies all the daily work with those stakeholders goes smoothly and makes project management easier.

Conclusions

The impact of corporate strengths/weaknesses on project management competencies was investigated in this study. According to the model presented in Figure 1. It was hypothesized that “corporate strengths/weaknesses”, so defined, impacts “project management competencies”. In order to test this hypothesis, a questionnaire survey was administered.

Based on the findings of the study, it can be stated that companies should adjust their resources and capabilities, their long term strategies and their relationships with other parties to better serve the needs of the individual projects. In the dynamic environment of the construction industry, companies have to behave farsighted in order to survive. Similar advice is of course good for companies in the area of information technology which is also very dynamic and project based. Tactical considerations which are short-term have to be complemented by long-term and strategic decisions. Finally, strong relationships should be developed with prospective clients, unions, and government.

Kari Jussila

Software product roadmapping in a volatile business environment

T. Suomalainen, O. Salo, P. Abrahamsson, and J. Similä, Journal of Systems and Software, volume 84, number 6, pages 958–975, 2011

Background

A product roadmap is an essential tool for businesses to see where they are going with their products and also as a company (p. 13). However, the authors feel that relatively little scientific knowledge on the subject exists in software development context (p. 1).

The study aims to increase the knowledge of software product roadmapping processes in both science and industry. Research is based on a framework that was constructed using existing literature. Using the framework following aspects were identified and studied: main stakeholders and their roles, organizing the roadmapping process, benefits and challenges and identifying the most critical phases. (p. 2)

Product roadmapping is defined as a process that gives a high-level view that links business aspects with requirements engineering, main purpose being to make sure that right products are released the right time and to improve cross-functional processes involved in creation of new products (p. 1).

Results

Before the study was executed an extensive literature review was done in order to create the research framework and the questionnaire used. After pilot testing the questionnaire, the research was done as a multiple-case study by sending the questionnaire via e-mail to over 600 companies that were selected using purposive sampling. Only 59 responses were received and seven of those had to be discarded, but the authors felt that the range of perspectives was wide enough. After the questionnaire nine interviews were conducted, representing eight companies, to gain more in-depth knowledge about the roadmapping processes used. (p. 9). Empirical findings were then analyzed and compared extensively with those suggested in existing literature.

Software product roadmapping as a process was found to consist of 3–6 distinct phases as suggested by literature. While different companies had different processes, in almost all cases similar tasks could be identified. Using the empirical data and the existing literature, a common framework for roadmapping is proposed that consists of five phases: capturing features, analyzing features, prioritizing features, roadmap validation and agreement, and change management. Roadmapping is done by a roadmapping team consisting of various stakeholders from different functions. (p. 17). Roadmapping team typically consists of the team members and the roadmap owner, who is responsible for all the administrative tasks. The team members typically are there only to give input (p. 15).

Stakeholders were divided into various categories according to their role in the company, where they represent different functions. According to respondents the most important

ones were the ones involved in product management, marketing and customer and partner representatives. The latter were the only external stakeholders involved in the roadmapping process. The size of the stakeholder team was clearly positively correlated with the company size. In majority of cases, the size of the roadmapping team was still less than 10 persons, however. In large companies the size could even go up to 100 persons, though not all these were necessarily involved in all the phases. (p. 10).

Capturing features, as the name suggests, involves capturing features to the roadmap. By far the most common way of doing this was simply gathering ideas over time, 65 % of respondents had used this method. This was followed by interviews and workshops. A host of other methods was also used, including prototypes, new standards, market research etc. Features were then analyzed using various methods including domain experts, interviews and feasibility studies. When analyzing features, following things were found to be most important: estimated cost, technical requirements and central use cases. Analysis was done through three steps: making sure everyone understands the feature, how it might be implanted (might involve prototypes) and cost analysis. (pp. 11–12)

Prioritizing features was in clear majority of cases done only using informal methods, in 69 % of responses. Both functional and non-functional requirements are prioritized at the same time using similar methods. Prioritization is typically done collectively by the roadmapping team but in some cases the customer or product owner made the decisions. Minor features were typically handled by the team and major ones might involve senior management from all the companies involved. (p. 12)

Two means of roadmap validation were found, reviews and customer feedback. Some interviewees found reviews the most efficient way of validating the roadmaps, especially when done by adequate amount of persons (p. 12). Sometimes roadmap validation was even a part of legal contract. When multiple companies were involved, the validation was usually done collaboratively. In the other case roadmap was validated retroactively by analyzing how business improved or didn't and analyzing customer feedback. This revealed whether the product had developed in a desirable direction. (p. 13).

Changes to the roadmap were most commonly caused by delays in implementation and changes in customer requirements. In the case of latter, customers usually brought up matters that hadn't been taken into account earlier. Changes were then analyzed as original features and approved by all involved parties. Major changes were usually related to schedules or adding/removing features. Decisions were typically done by the roadmapping team but in case of customer funded project, the customer did them. (p. 13).

Most important phases in the process according to the data clearly are prioritizing and capturing features. Change management was clearly considered least important, since good change management won't help if other phases have gone wrong. The difficulty of phases was also queried. Medium and large companies found feature prioritization to be the hardest one, while small companies thought it was validation and agreement. Product lifetime also affected the perceived difficulty; those with product lifetimes of less than six years had difficulties in all but change management, while companies with longer product lifetimes additionally didn't have problems in roadmap validation. (p. 14).

However, what was commonly found difficult in all phases was related to

communication. Getting the right input, acquiring customer and competitor information, communication with customers and in large companies, consolidating different viewpoints, prioritizing customer requirements and sharing information were all considered somehow problematic. Problem is partially caused by the fact that people that have access to the roadmap have to be verified, since leaks could be potentially very harmful to the business. (pp. 14–15).

Conclusions

Respondents felt that roadmapping is a means of structuring and arranging product development, that provides a high-level understanding of company strategy. It also guides product development by saying what is done and when. Roadmapping also improves predictability and reduces surprises. Roadmap helps in resource allocation as they can be allocated to most profitable projects and by incorporating customers' needs it can give a competitive advantage. It can also be a powerful vehicle for communicating the company strategy within and without the company. (p. 13).

In the industry, companies can use the findings to focus their work on the critical parts of the process. From a scientific perspective the body of knowledge regarding the subject is increased, in addition to the research framework constructed. Study also provides a basis for studying the roadmapping process in the context of agile software development methods. (p. 17).

While the paper is comprehensive and methodically covers all the areas defined in the research objectives, it is somewhat lacking in real world examples and direct quotations from interviewees are used too sparingly in my opinion.

Ville Autio

User and developer common knowledge: Effect on the success of information system development projects

D. Tesch, M. G. Sobol, G. Klein, and J. J. Jiang, International Journal of Project Management, volume 27, issue, 7, pages, 657-664, 2009

Johdanto

Artikkelissa tutkitaan sekä tulkitaan ohjelmistokehittäjien ja asiakkaiden, tuotteen käyttäjien yhteisen tiedon ja taidon vaikutuksia ohjelmistoprojektien onnistumisiin. Erityisesti tutkinnassa on keskitytty näiden kahden tahon yhteisten ongelmanratkaisukykyjen ja käytäntöjen korrelointia projektien onnistumisen kanssa.

Vastaavaan erityisalueen tutkimuksissa päädytään yleensä kahteen teoriaan, vahvan suhdetoiminnan (Strength-of-tie relationships theory) ja tiedon yhtenäistämisen teoriaan (knowledge integration theory). Vahvan suhdetoiminnan teoria korostaa kehittäjien ja asiakkaiden päivittäistä yhteistyötä ohjelmiston luonnissa. Erityisesti teoria korostaa yhdessä tehdyn ongelmanratkaisutyön arvoa ja osuutta projektien onnistumisessa. Tiedon yhtenäistämisen teoria korostaa enemmän yhteisen, yhtenevän tietopääoman määrän merkitystä projektien laadulle.

Tulokset

Tekijöillä oli tutkimukseen alkaessa kaksi hypoteesia;

1. Asiakkaiden ja kehittäjien välinen ongelmanratkaisutyö vaikuttaa/ei vaikuta asiakkaiden sovelluskehitys-tuntemuksen määrän vaikutusvoimakkuuteen projektin lopputulokseen.
2. Asiakkaiden ja kehittäjien välinen ongelmanratkaisutyö vaikuttaa/ei vaikuta kehittäjien toimialatuntemuksen määrän vaikutusvoimakkuuteen projektin lopputulokseen.

Tutkimus suoritettiin kyselytutkimuksena ammattilaisille sovelluskehitys - projektipäälliköille. Yhteistä kyselyyn osallistuneille oli osallisuus projektipäälliköille tarkoitettussa SIG (special interest group) -ryhmässä. Kysymysten vastaanottajat valittiin SIG-ryhmän jäsenistä satunnaisotannalla. Vastauksia palautettiin 171 kappaletta. Kysymyspatterissa kysyttiin erilaisten tilanteiden ja väittämien oikeellisuutta ja esiintymistä todellisissa projekteissa. Vastaajat arvioivat jokaisen kohdan asteikolla 1-5. Hyväksyttäviä vastauksia saatiin 167 kpl. Vastaukset ja niiden suhde hypoteeseihin käsiteltiin PLS (Partial Least Squares) -menetelmällä.

Tutkimus osoitti, että sekä kehittäjien toimialatuntemus että asiakkaiden sovellusprojektiymmärrys näyttelevät suurta roolia projektien onnistumisessa. Lisäksi, mitä suurempi määrä projektissa oli asiakkaiden ja kehittäjien yhdessä suorittamaa ongelmanratkaisutyötä, sitä suurempi yhteisen tietotaidon määrän vaikutus oli projektin onnistumiselle.

Tulokset osoittavat, että jo projekteja käynnistäessä tulisi kiinnittää huomiota sovelluskehittäjien ja asiakkaiden yhteisen tietotaidon määrään. Mikäli yhteisen

tietotaidon määrää ei kuitenkaan saada halutulle tasolle projektia muodostaessa, tämä korostaa tarvetta projektin aikana suoritettavalle yhteiselle ongelmanratkaisutyölle.

Lisäksi tulokset korostavat, että yhteisen tietotaidon harjoittelu ja opiskelu koko projektin elinkaaren ajan olisivat tarpeen enemmissä määrin mitä yleensä harjoitetaan.

Tuloksia arvioidessa tulee ottaa huomioon, että kehittäjät ja asiakkaat näkevät yleensä projektien onnistumisen asteen eri lailla. Arvioitavat seikat ja näkökulmat eroavat toisistaan. Lisäksi tulee huomioida, että esim. viestintätaitoja ei otettu tässä tutkimuksessa huomioon.

Johtopäätökset

Ohjelmistokehitysprojektien Akilleen kantapää on usein se, etteivät kehittäjien ja asiakkaiden tiedot kohtaa. Asiakkaat osaavat sen alan tai työn, johon uutta ohjelmistoa rakennetaan. Mutta heillä ei ole ymmärrystä eikä osaamista ohjelmistotuotannosta tai kehitysprojektien onnistumiseen vaikuttavista projektillisistä tekijöistä. Ohjelmoijat osaavat teknisesti rakentaa ohjelmistoja, mutta eivät usein tunne itse ydintoimintaa ja siihen liittyviä yksityiskohtia. Usein käy niin, etteivät asiakkaiden sekä kehittäjien tarpeet, toiveet ja tavoitteet kohtaa optimaalisella tavalla. on tyypillistä, että sovelluskehittäjien mielestä tuote on projektin päättyessä riittävän tasoinen, kun taas asiakkaat havahtuvat ongelmiin vasta käytön jo alettua Tässä vaiheessa uudelleen aloitettava “paikkaava” kehitystyö on sopimusteknisesti usein jo uusi, uudelleenhinnoiteltu projektinsa.

Jussi kivinen

Agile and Kanban in coordination

R. Polk, in Proceedings of Agile Conference 2011, pages 263-268, 2011

Johdanto

Tämä artikkeli kertoo 18 henkilön kehittäjätiimin organisoinnista WMS Gaming-yrityksessä. Työ koostuu 60-prosenttisesti laajamittaisista järjestelmän ominaisuuksista ja loppuosa työstä koostuu pienimuotoisista muutoksista sekä bugikorjauksista. Ennen muutoksia tiimi käytti vesiputousmallia ja muutosten jälkeen käytössä oli yksinkertainen ja toimiva ketterä malli.

Muutosten jälkeen oli kolme ryhmää: yksi iteratiivinen ryhmä, joka hallitsi isompia kokonaisuuksia, yksi Kanban-ryhmä, joka hallitsi pienempiä kokonaisuuksia sekä tuotteen omistaja-ryhmä, joka hallitsi työnjakoa ryhmien välillä.

Kanban tarkoittaa taulua, joka jaetaan alueisiin kuten valmiit kehitettäväksi, kehitys, testaus ja valmis. Jokaisella alueella voi olla arvo, joka merkkää, kuinka monta tehtävää voi olla kyseisessä vaiheessa kerrallaan.

Tulokset

Alussa tiimi yritettiin jakaa kolmeen tiimiin, jotka työstivät samaa tuotetta (p. 263). Ensimmäisten kuukausien aikana selvisi, että tämä tapa ei toiminut kovin hyvin. Yksittäiset tehtävät saattoivat jakautua monelle ryhmälle ja yksittäisten kehittäjien piti hyppiä tiimien välillä että he saisivat työtä tehtyä.

Tämän jälkeen siirryttiin yhteen tiimiin, mutta tästä syntyi se ongelma, että 3-5 jäsentä ei onnistunut tekemään töitään ja he pystyivät piileskelemään suuren ryhmärakenteen takana (p. 263).

Muutaman kuukauden keskustelun jälkeen hallintotiimi päätyi luomaan kaksi ryhmää (p. 264). Ensimmäinen ryhmä koostui yhdeksästä hengestä, jotka aikaisemmin olivat keskittyneet suurempien ominaisuuksien tekemiseen. Tämä ryhmä otti käyttöön iteratiivisen kehitysmallin. Toinen ryhmä oli kuuden hengen Kanban-tiimi, joka keskittyi pienempiin ja paremmin määriteltyihin ominaisuuksiin. Tämä tiimi koostui pääosin henkilöistä, jotka onnistuivat välttämään töitä edellisen järjestelyn aikana.

Managerit olettivat, että Kanban-ryhmään laitettut henkilöt eivät saaneet tarpeeksi hyvää rakennetta tukemaan heitä, mistä syystä heidät laitettiin tähän pienempään ryhmään. Ryhmä otti Kanbanin käyttöön auttamaan työn hallitsemista.

WMSsän metodiin vaikutti vahvasti Scrum ja eXtreme Programming. He päätyivät seuraaviin kahdeksaan käytäntöön (p. 264); käyttäjäkertomukset, hyväksyntätestit, burn down -kaaviot, projektitaulut, päivittäiset Stand-Up -kokoukset, automatisoidut yksikkötestit ja jatkuva integraatio. Kanban-ryhmältä otettiin pois jatkuva integraatio sekä burn down -kaaviot, ja taulu korvattiin Kanban-työkalulla (p. 265).

Koska ryhmä oli jo tottunut projektitaulun käyttöön, Kanbaniin siirtyminen oli helppoa (p. 264). Koska ryhmässä ei ollut testaajia, ryhmän jäsenet vuorottelivat niin, että yksi henkilö teki testejä valmiiksi tehdyille koodille. Tämä henkilö myös koordinoi näiden ominaisuuksien toimituksen Product Assurance tiimille.

Muutoksen alkuvaiheilla luotiin tuotteen omistaja-ryhmä, joka koostui yhdestä managerista ja kahdesta kehittäjästä. Tämän ryhmän tarkoituksena oli ylläpitää säännöllisiä tapaamisia ja jakaa uudet tehtävät ryhmien välille. Ryhmä ylläpiti myös Kanban-tilin tehtävien tärkeysjärjestystä. Heillä ei ollut lupaa koskea mihinkään muuhun kuin tehtävien tärkeysjärjestykseen, paitsi pelkästään hätätilanteissa.

Kehitys seurasi iteratiivisen tiimin työskentelyä. Heillä oli 2-3 tehtävää kerrallaan. Iteraatiot kestivät kaksi viikkoa ja tätä ylläpidettiin projektitaululla. Iteraation suunnittelukokous pidettiin aina iteraation viimeisenä perjantaina.

Kanban tiimin taulu oli iteratiivisen tiimin taulun vieressä. Kanbanin jonoiksi valittiin loppujen lopuksi seuraavat; 1. Odotusjono, 2. Tarinat, maksimimäärä tehtäviä 5, 3. Kehitys, maksimimäärä tehtäviä 5, 4. Kehitys valmis, maksimimäärä tehtäviä 3, 5. Varmistus ja hyväksyntä, maksimimäärä tehtäviä 4, 7. Valmis ja 8. Yhdistetty / Otettu käyttöön.

Kanban-tiimissä huomattiin se, että kun taululle otettiin liian isoja palasia, nämä jäivät taululle hidastamaan muuta kehitystä (p. 266). Päätettiin, että suurin sallittu tehtävän koko on 8 pistettä.

Tuotteen omistaja -ryhmä piti päivittäiset palaverit aamulla kummankin kehitysryhmän kanssa niin, että aluksi palaveri käytiin iteratiivisen ryhmän kanssa ja sitten Kanban-ryhmän kanssa (p. 266). Kummankin palaverit kestivät 15 minuuttia. Tämän avulla tuotteen omistaja -ryhmä pystyi pysymään kärryillä siitä, mitä tapahtuu.

Tämä ryhmä koordinoi 99-prosenttisesti kaikkia pyyntöjä (p. 266). Jäljelle jäävä yksi prosentti koostui bugikorjauksista ja viime hetken muutoksista ominaisuuksiin. Nämä hoidettiin niin, että korvattiin jotain iteratiivisen tiimin taululta tai lisättiin Kanban-jonoon.

Työ jaettiin niin että tuotteen omistaja -ryhmä jakaa ja hajoittaa työn hallittaviin erikokosiin paloihin. Tärkeimmät tehtävät laitetaan kummankin ryhmän jonon eteen. Kanban-ryhmä otti seuraavaksi pienemmät ja helpommat tehtävät. Sitten iteraatioryhmä otti tärkeimmät toiminnallisuudet kehitykseen. Näin työ saatiin helposti jaettua.

Kun työnteon nopeutta mitattiin, Kanban-ryhmän tehokkuus alkoi kasvamaan kahden ensimmäisen kuun jälkeen (p. 267). Tällöin otettiin myös käyttöön mittari, josta yksittäiset henkilöt pystyivät seuraamaan omaa tehokkuuttaan. Seitsemässä kuukaudessa Kanban-ryhmä pääsi tehokkuudessa suunnilleen samalle tasolle, missä iteratiivinen ryhmä oli. Tänä aikana ryhmän motivaatio ja energia nousi mittaamattoman paljon.

Yhteenveto

Kuten tässä artikkelissa kerrottiin, Kanbanin sekoittaminen muihin ketteriin menetelmiin ei ole ainut mahdollisuus kun yritetään parantaa ryhmien tuottavuutta. Myös eri tavalla toimivien ryhmien yhdistäminen on mahdollista. Tämä tulee tehdä tapauskohtaisesti.

Tämän artikkelin esimerkki kuulosti mielestäni erinomaiselta. Hyvin toimivat tiimin jäsenet laitettiin yhteen tiimiin, joka toimi perinteisellä tavalla. Heidän ei tarvinnut opetella mitään ja he saivat jatkaa kehitystä samalla kun huonommin toimivat jäsenet laitettiin toiseen tiimiin, jolle annettiin aiempaa selkeämpi rakenne. Tätä suositellaan myös muille samantapaisille suurille kehitystiimeille, jotka kehittävät yhdessä yhtä tuotetta (p. 268).

Toni Järviluoto

Human resources management main role in information technology project management

H. Tohidi, Procedia Computer Science, volume 3, pages 925-929, 2010

Johdanto

Erilaisten projektienhallinnan menetelmä tapojen suosio on ollut jatkuvasti nousussa, kun projekti- ja päätkätyöt määrittävät yhä enemmän nykyistä työskulttuuria. Näin on erityisesti IT -alalla, mutta näyttäisi siltä, että jostain syystä IT – projektit eivät onnistu yhtä usein kuin muiden alojen projektit. USA:ssa tehdyssä tutkimuksessa todettiin, että noin 30 % IT-projekteista keskeytetään ja yli puolet projekteista ylittää budjetin lähes kaksinkertaisena (Chaos, “Application Project Failure and Success”, p.89 Standish Group, 1995).

Jotta projektien onnistumisprosenttia ja niiden toteutumistehokkuutta voitaisiin nostaa IT -alalla, tarvitaan enemmän tietoa siitä, mitkä tekijät vaikuttavat projektin onnistumiseen tai epäonnistumiseen. Nykyisellään kehitys on niin nopeata IT alalla, että monet asiantuntijat yrittävät asettaa rajoja sille sekä arvioida ja mitata eri tavoin sen vaikutuksia tulevaisuuden kannalta.

Tietojärjestelmäprojekti on yleensä tyypiltään kehitysprojekti. Tällainen tutkimusprojekti kuitenkin muistuttaa tavallisen projektityön yleisiä piirteitä. Tietojärjestelmäprojektissa on asetettu tavoitteeksi useimmiten jo olemassa olevan toiminnan parantaminen tai kehittäminen. Tavoitteena voi olla toisaalta myös kokonaan uuden toiminnan luominen.

Varsinkin yksityiskohtaisen suunnitelman määrittely projektin alussa on lähes mahdotonta. Työn edetessä kuva lopputuotoksesta kuitenkin selvenee. IT projekteissa suunnitelman muuttuminen ja mukautuminen on hyvin paljon yleisempää kuin yleensä.

Koska projektia käytetään työmuotona tietyn päämäärän saavuttamiseksi, projekteja voidaan käyttää monilla eri IT soveltavilla aloilla. Tässä tutkimuksessa on esitetty valtion viraston systeemyötä henkilöstöhallinnan näkökulmasta. Eri alojen projektit voivat kuulostaa pääpiirteiltään varsin samanlaisilta ja muistuttavat toisiaan, mutta sisällöltään ja toteutukseltaan ne voivat olla hyvin erilaisia.

Projektille ominaisia piirteitä ovat aikataulu, budjetti ja ainutlaatuisuus. Tietojärjestelmäprojekti muistuttaa tavallista projektia. Eroavaisuuksia tavalliseen projektiin on abstrakti lopputuotos ja teknologian luoma epävarmuus. Projektin onnistumista arvioidaan erilaisin keinoin. Suuntaa antavan arvion onnistumiselle voi arvioida kolmen tekijän kautta. Jos projekti täyttää ajan, budjetin ja laadun puitteissa asetetut kriteerit, niin se voidaan mieltää alustavasti onnistuneeksi. Projektipäällikön ominaisuuksia ja kykyjä käsitellään kompetenssien kautta. Kompetenssit ovat henkilöstön luonteenpiirteitä ja ominaisuuksia.

Ensimmäinen toimintaohjelma henkilöstöhallinnossa on rekrytointiprosessi eli toimittaa IT työvoimaa projektin vaatimuksiin. Jos tämä vaihe ei ole tehty oikein, aiheuttaa

epäsopivien henkilöiden valinta projektin edetessä piileviä ja heti havaittavia ongelmia. IT organisaatioiden henkilöstön koulutustausta on erittäin tärkeä merkitys ja pääoma yritykselle. Henkilöt, jotka ovat itsenäisiä, luotettavia ja vastuullisia sopivat IT hankkeisiin. Monet johtavat tietojärjestelmä-konsultit uskovat siihen ajatukseen, että 80 prosenttia projektinhallinnasta, on viestintää. He eivät voi kaikki olla väärässä. Tehokas viestintä on yksi keskeinen edellytys mihinkä tahansa projektille. Vaikka hankkeella olisi vahva tekninen perusta, johtaa puutteellinen kommunikointi projektin edetessä usein sen epäonnistumiseen. Joten johtajalla tulisi olla tehokkaita menetelmiä saamaan aikaan tehokas viestintä. Sen toimintaa tulisi suunnitella myös henkilöstöä rekrytoitaessa sekä tällöin kiinnittää huomiota heidän persoonallisuutensa sopivuuteen IT tehtävissä.

Henkilöstöjohtamista voidaan pitää yhtenä tietojärjestelmäprojektin kriittisenä menestystekijänä. Henkilöstötutkimus on lähtökohdiltaan organisaation henkilöstön tyytyväisyyttä, asenteita, työssäjaksamista tai ammattitaitoa mittaava ja analysoiva tutkimus. Henkilöstötutkimuksen avulla pyritään tyypillisesti toteuttamaan työtyytyväisyys- ja sekä työhyvinvointi barometrin luominen. IT organisaation onnistumisen ehtona on johtotyöskentelyn tehostaminen ja henkilöstölähtöisen motivointijärjestelmän kehittäminen sekä organisaation sisäisen toiminnan kokonaisvaltainen kehittäminen. Tietojen analysointiin käytettiin tilastollisia menetelmiä, kuten Friedman, Pearson, Keydo ja Spearman, ja tilastollisesti merkittävien tulosten aikaan saamiseksi. Tärkeimmiksi havainnot osoittavat henkilöstön kannalta olevan yksilölliset tekijät, kuten luonteen laatu ja sosiaalisuus. Kun taas tekijät henkilöstön työn onnistumisen, ja lopulta koko organisaation kannalta ovat henkilöstön tehokkuus ja suorituskyky. Työntekijöiden vastuuntunto ja sitoutuminen työhön sekä tietotaidontaso tietotekniikan alalla tunnustettiin olevan yksi tärkeimmistä tekijöistä parantamaan suorituskykyä henkilöstön yksilötasolla tiimeissä.

Myös ottaen huomioon ammatilliset tekijät, sekä tarvittava spesifinen koulutuksen saaminen tehtäviin mahdollisuus työpaikalla, oli eräs ratkaisevista tekijöistä IT yrityksissä. Seuraavaksi huomio palkkauksessa ja sen oikeudenmukaisuudessa sekä riittävyudessa olivat huomattavia tekijöitä tutkittavissa työpaikoissa IT yrityksissä. Seuraavassa luettelossa on joitakin näistä tekijöistä esitetty tärkeysjärjestyksessä (taulukko 1).

Tulokset

Tutkimustyössä alalla tietotekniikasta vastaava henkilöstö oli valtion virasto, joka toimi aktiivisesti osallistumalla tietotekniikan hankkeisiin. Täällä kiinnitettiin huomiota tehokkuuteen vaikuttaviin tekijöihin, jotka vaikuttavat parantavasti henkilöstön suorituskykyyn. Näihin hankkeisiin ja tähän tutkimustarkoitukseen otettiin tilastoja ja priorisointiin tilastojen tekijöitä, jotta saatiin esille tärkein informaatio henkilöstön näkökulmasta. Tämä tutkimus perustui kysely menetelmään (survey), jossa otoskoko oli tutkimuksessa yksinkertainen satunnaisotanta ilman ennakoasetuksia projektin henkilöstöstä. Kaavassa käytetyn tilastollisen muuttujan koko oli (Cochrane database) tietokannasta otetun näyttöiden määrä, joka oli 45 valittua IT projektin henkilöä valtion virastossa. Voitiin määrittää valmistellun kyselylomakkeen avulla tärkeimmät henkilöstöresurssien työskentelytehoon vaikuttavat tekijät. Tehtiin lopuksi tilastoanalyysi, joka perustui muodolliseen survey tyyppiseen lineaariseen malliin.

Menetelmä on yleisesti käytetty ja sen luotettavuuden vahvistivat asiantuntijat, joiden antama luotettavuus Cronbach alfa kertoimen mukaan oli 0.86.

Taulukko 1. Työyhteisön henkilöstöä motivoivien tekijöiden merkitys.

| Yksilölliset tekijät | Työntekoa motivoivat tekijät | Organisatoriset tekijät |
|---|--|--|
| Vastuullisuus ja vastuuntunto työssä | Koulutus ja tutkimustarve | Oikeudenmukainen palkkaus ja riittävät korvaukset |
| IT-taidot ja osaaminen | Organisaation kehittämistarpeen huomiointi | Oikea työympäristö |
| Intohimo ja kiinnostus tehtäviä kohtaan | Huomioida työn vaativuus ja tietotaitotaso | Huomioida IT johtaja ja asiantuntija tehtävien erilaiset vaatimustasot |
| Oikea yhteispelihenki | Ottaa huomioon muiden työn kunnioitus ja arvostus | Kiinnittää huomiota työsuhteturvaan ja työehtoihin |
| Arvioida muutostarpeet IT työssä oikein | Välttävät konfliktit rooleissa, tehtävissä ja vastuissa | Luoda edellytykset itsensä kehittämiselle |
| Tietotaidon ja oman osaamisen välittäminen muille työyhteisössä | Ryhmätyövalmiudet ja työn jakamiseen tiimityöskentelyssä | Mahdollisuudet stressin lieventämiseen ja välttämiseen |
| Intohimo ja kiinnostus omaa työpaikkaa kohtaan | Vaikutusmahdollisuus päätöksentekoon työssä | Enemmän arvostusta henkilöstön kokemukselle ja osaamiselle työssä |
| Työkokemus ja ammatillinen tausta | Välttää yksitoikkoinen ja monotoninen työtehtävä | IT messut ja tutustumistilaisuudet |

Johtopäätökset

IT projektinhallinnalle ei yleensä ole määrättyä reseptiä ja et voi kehittää kattavaa suunnitelmaa kaikille hankkeille tällä alalla. Kaikkia osapuolia tyydyttävä IT projektin hallinta ei ole mahdollista haastattelujen valossa niiden mielestä, jotka ovat saaneet paljon ja hyvin erilaisista hankkeista kokemusta IT alalla. Lähes kaikilla organisaatioilla on omat ongelmansa ja vaikeutensa. Mutta meidän on oltava huolellisia suunnitteluvaiheessa sillä on hyvin yleisesti olemassa useita ongelmia IT projekteissa. Projektin hallinta on mahdollista vain huomioiden, että eri yritysten käytännöt ja toimintaympäristö ovat hyvin erilaisia. Ei ole yhtä oikeaa toimintatapaa tai ratkaisua eri IT projekteissa, sillä niihin liittyvät ongelmat ja tavoitteet käyttäjätasolla ovat eri organisaatioissa hyvin erilaisia. Tämän tutkimuksen tulokset osoittivat, että tulee kiinnittää heti projektin alussa huomiota oikean henkilöstön valintaan organisaatiossa ja henkilöstöhallintaan on yksi tärkeimmistä pilareista onnistuneelle IT johtamiselle. IT johtamisen kyvyttömyydestä aiheutuu suurin syy IT projektien epäonnistumiseen tutkituissa hankkeissa.

Jyrki Hemming

The agile requirements refinery: Applying SCRUM principles to software product management

K. Vlaanderen, S. Jansen, S. Brinkkemper, and E. Jaspers, Information and Software Technology, volume 53, issue 1, pages 58-70, 2011

Johdanto

Ketteriä menetelmiä on käytetty ohjelmistotuotantoprojekteissa jo vuodesta 1957 iteratiivisten menetelmien muodossa mutta nykyään tunnetut ketterät menetelmät ovat tulleet muotiin vasta 1990 –luvun puolenvälin paikkeilla. Vuonna 2001 julkaistiin ‘Agile Manifesto’ joka piti sisällään nykypäivän ketterät menetelmät kuten esimerkiksi SCRUM.

SCRUM, kuten muutkin ketterät menetelmät ovat keskittyneet enemmän tuotettuun ja tuotettavaan sovellukseen kuin sen dokumentaatioon, työkaluihin ja prosesseihin. Tämän vuoksi ketterät menetelmät sopeutuvat paremmin aloihin ja projekteihin, joissa ei välttämättä ole heti kaikkia vaatimusmäärittelyjä valmiina. Ketterien menetelmien ideana onkin ottaa ideoita ja vaatimuksia huomioon hyvin nopealla reagointi ajalla, sprinteissä, toisinkuin muissa menetelmissä, missä vaatimuksien muuttaminen on vaikeampaa ja hitaampaa. Ketterät menetelmät ovat sopeutuneet hyvin ohjelmistotuotantoprojekteihin ja siksi onkin luontevaa että seuraava kehitysaskel on sopeuttaa kyseiset menetelmät projektin managerointiin tai johtamiseen. Ketteristä menetelmistä ohjelmisto johtamisen osalta ei ole paljoa tutkimusta tehty, muuta kuin Pichler et al. sekä Vähäniitty ja Rautiainen osalta. (p. 59)

SCRUM menetelmää käyttävä projektin johtaminen pitää sisällään samanlailla kuin normaalissa SCRUM:ssa työlisteriä joka pitää sisällään vaatimusmäärittelyn osia. Nämä osat ovat nimetty ‘Vision’, ‘Theme’, ‘Concept’ ja ‘Requirement’. Sprintissä näitä osia hiotaan ja pilkotaan pienemmäksi ja pienemmäksi kunnes ne ovat tasolla jolla ne voidaan todeta vaatimusmäärittelyksi. Tämän jälkeen ne hyväksytetään ja mikäli ne on hyväksytyt ne siirtyvät esimerkiksi tuotantotiimin työlisteriin.

Tulokset

Artikkelissa tutkitaan tapaustutkimuksen kautta miten SCRUM menetelmää voidaan soveltaa projektin johtamisessa. Tapaustutkimus tehdään hollantilaisessa ohjelmistoyrityksessä Platon:ssa, jossa kyseistä menetelmää on käytetty jo kahden vuoden ajan hyvin tuloksin. Tapaustutkimuksen kaksi pääkysymystä olivat että miten SCRUM menetelmää voidaan käyttää ohjelmistoprojektin johtamisessa ja miten sen menetelmän tuominen vaikuttaa johtamiseen. Ensimmäiseen kysymykseen on jo vastattu sillä miten SCRUM on muunneltu toimimaan projektin johtamisen osalta. Toiseen kysymykseen vastauksia haetaan kahdella eri tavalla: haastatteluilla sekä dokumenttien tarkastelulla.

Jotta empiirinen tutkimus olisi oikeellista ja pätevää, on tutkimukseen otettu mukaan neljä eri kriteeriä, joiden mukaan tutkimus on rakennettu. Nämä kriteerit ovat ‘Construct’, ‘Internal’, ‘External’ sekä ‘Reliability’. Construct kriteeri pitää sisällään sen,

että tutkimustiimillä on selvä ja oikea käsitys ketteristä kehitysmenetelmistä ja SCRUM:sta niin sen käytännön teoriasta kuin terminologiasta. Internal-kriteeri pitää sisällään sen, että informaatio on samaa vaikka se on lähtöisin eri lähteistä. Tästä pidettiin tutkimuksessa huolta ristiin haastatteluilla sekä dokumenttien ristiinvertailulla. External kriteeri piti sisällään tiedon siitä, että tämä tutkimus on tehty yhdestä yrityksestä eikä se ole yleistettävissä muihin yrityksiin. Reliability kriteeri piti sisällään, että vaikka tutkimus tehtäisiin uudestaan, saataisiin samat tai samankaltaiset tulokset, sillä protokollaa ja tapaustudkimuksen rakennetta on noudatettu tarkasti.

Tutkimuksessa kävi ilmi, että Platonin projektin työlistaan ilmestyi muiden “normaalien” tehtävien lisäksi useita projektin johtamiseen ja hallintaan sekä seuraaviin sprintteihin liittyviä tehtäviä kuten esimerkiksi sprintin alku ja lopputarkastelu. Vaikka tämä jo itsessään on hyvä tulos siitä, että SCRUM menetelmää voidaan käyttää projektin johtamiseen, tutkimuksessa silti edettiin pidemmälle tarkasteluissa ja tutkittiin muunmuassa roolitusta projektissa ja miten SCRUM menetelmä paransi työtehoa tältä osin.

Tutkimuksessa käytiin lävitse yhden ‘Theme’:n eli ajatuksen kehittyminen ja miten se kulki lävitse kehitysmallin prosessit. Kyseisen ‘Ylläpito suunnitelma’ -ajatuksen elosykli oli noin seitsemän kuukautta jonka aikana se kävi lävitse useita sprinttejä vähitellen muuttuen ajatuksesta konseptiksi ja konseptista useaksi vaatimusmäärittelyksi, jotka sittemmin toteutettiin osana projektia.

Toisena tarkempaan tarkasteluna tutkimuksessa käytiin lävitse ‘Platon Lite’. Platon Lite:n elosykli kesti noin yksitoista kuukautta, jonka jälkeen se hävisi työlistasta. Platon Lite idea siis hylättiin. Vaikka Platon Lite hylättiin, se antoi kaksi erinomaista tutkimus tulosta. Ensimmäinen oli, että kaikki ominaisuudet eivät ala ‘Theme’ eli ajatus tasolla, vaan voivat tulla työlistalle jo esimerkiksi konseptina. Toinen tulos oli se, että tehtävät palaavat työlistan etupäähän aika ajoin, joka indikoi, että SCRUM menetelmä toimii johtamistasolla antaen lisää selkeyttä ja rakennetta johtamisen suunnittelussa ja toteuttamisessa.

Yhteenveto

Artikkelin mukaan SCRUM mentelmän implementoimisen myötä Platon on saanut arvokasta tietoa ja taitoa SCRUM:n käytöstä projektin johtamisessa. Myös tiedemaailma on mielestäni saanut valtavan annoksen lisätietoa SCRUM:n käytännön mahdollisuuksista. Itse henkilökohtaisesti en ole ajatellut enkä tajunnut ketterien menetelmien potentiaalista mahdollisuutta nousta tälle tasolle suunnittelumenetelminä. Artikkelin oli mielestäni erittäin mielenkiintoinen ja toi uusia näkökulmia projektin johtamiseen.

Mikäli olisin löytänyt artikkelin hieman aikaisemmin, esimerkiksi viime vuonna, olisi tämän hetkinen Software project management –kurssi ollut osaltani täysin erilainen. Ajatus “SCRUM:sta SCRUM:n sisällä” on kiehtova.

Pekka Ihalainen

Software process improvement barriers: A cross-cultural comparison

M. Niazi, M.A. Babar, and J.M. Verne, Information and Software Technology, volume 52, issue 11, pages 1204-1216, 2010

Johdanto

Ohjelmisto-prosessien parantamisessa (Software Process Improvement - SPI) on kyse erilaisista toimintamalleista, joiden avulla ohjelmisto-organisaatiot pyrkivät kehittämään korkealaatuisia ohjelmistoja entistä tehokkaammin. Aiemmat tutkimukset ovat kuitenkin osoittaneet, että ohjelmistoja kehittävätkin organisaatiot eivät ole aina halukkaita käyttämään ohjelmisto-prosessien parantamiseen tähtääviä toimintamalleja, koska näiden toteuttaminen vaatii pitkää sitoutumista ja tämä voi siten olla kallista organisaatioille. Pienillä ja keskisuurilla organisaatioilla on myös herännyt kysymyksiä näiden toimintamallien merkityksistä ja käyttömahdollisuuksista käytännössä.

Ohjelmistoyhtiöt siirtyvät enenevässä määrin globaaliseen ohjelmistokehitykseen, jossa ohjelmistokehityksen osa-alueita ulkoistetaan säästöjen saamiseksi esimerkiksi kehittyviin maihin. Monet ohjelmistoja myyvät organisaatiot pyrkivät ottamaan käyttöön ohjelmisto-prosessien parantamiseen tähtääviä toimintamalleja kilpaillakseen paremmin kansainvälisillä ulkoistamismarkkinoilla, jossa näiden toimintamallien käyttö voi merkittävästi parantaa organisaation mahdollisuuksia saada uusia sopimuksia.

Ohjelmistokehityksen käytännöistä ja ohjelmisto-prosessien parantamiseen tähtäävien toimintamallien käyttämisestä on olemassa paljon kirjallisuutta, joka koskee kehittyneitä maita ja perinteistä ulkoistamisen kohdetta Intiaa. Samalla on kuitenkin olemassa hyvin vähän empiirisiä tutkimuksia, jossa selvitetään miksi organisaatiot eivät käytä näitä ohjelmisto-prosessien parantamiseen tähtääviä toimintamalleja ja mitkä ovat olennaiset esteet (barriers) ohjelmisto-prosessien parantamiseen tähtäävien aloitteiden toteutumiselle.

Vietnamista on kehittymässä yksi merkittävimmistä ulkoistamisen kohteista ohjelmistoalalla. Tutkimuksessa keskitytään vertailemaan kehittyneen maan, Australian, ja kehittyvän maan, Vietnamin, eroja ja yhtäläisyyksiä ohjelmistokehityksen käytännöissä ohjelmisto-prosessien parantamiseen tähtäävien aloitteiden näkökulmasta. Tutkimuksessa keskitytään erityisesti kahteen kysymykseen:

"Millä tekijöillä Vietnamilaisien ammatinharjoittajien mukaan on negatiivinen vaikutus ohjelmisto-prosessien parantamiseen tähtäävien aloitteiden toteutumisessa?" ja

"Ovatko nämä tekijät samoja kehittyneissä ja kehittyvissä maissa?"

Tulokset

Tutkimus toteutettiin empiirisellä tutkimuksella, jossa Vietnamilaisien ohjelmistoammattinharjoittajien näkemyksiä ohjelmisto-prosessien parantamiseen tähtäävien aloitteiden toteutumisen esteistä verrattiin aiempiin tutkimuksiin, joissa selvitettiin Australialaisten ohjelmistoammattinharjoittajien näkemyksiä samasta asiasta.

Osana tutkimusta haastateltiin 34 Australialaista ohjelmistoalan ammatinharjoittajaa sekä 23 Vietnamilaista ammatinharjoittajaa. Haastattelujen lisäksi tutkimukseen osallistuneet ammatinharjoittajat vastasivat kyselyyn, jossa ammatinharjoittajilta pyydettiin listaamaan niitä seikkoja, jotka heidän mukaansa ovat esteitä ohjelmisto-prosessien parantamiseen tähtäävien aloitteiden toteutumisessa ja arvioivan kuinka vakavia nämä seikat ovat.

Tutkimuksen päämääränä on selvittää miten ohjelmisto-prosessien parantamiseen tähtäävät aloitteet tulisi toteuttaa, jotta ohjelmisto-organisaatiot voisivat paremmin toteuttaa ohjelmisto-prosessien parantamiseen tähtäävien toimien käyttöönoton. Erityisesti on pyritty selvittämään ne organisaatiolliset seikat, jotka voivat haitata ohjelmisto-prosessien parantamisen toteutumista ja miten ohjelmisto-organisaatiot voivat edistää ohjelmisto-prosessien parantamista ottamalla nämä seikat huomioon.

Tutkimuksen mukaan suurimmat esteet ohjelmisto-prosessien parantamisen toteutumiselle Vietnaminissa ovat ohjelmistoprojektien johtamisen puute, resurssien puute, sponsorien puute, henkilöstön kokemuksen tai tietotaidon puute sekä ohjelmisto-prosessien parantamisen tietoisuuden puute. Tutkimuksessa tulee ilmi eroja ja yhtäläisyyksiä Australialaisten ja Vietnamilaiden ammatinharjoittajien välillä niistä seikoista, joita pidetään suurimpina esteinä ohjelmisto-prosessien parantamisen toteutumiselle. Myös Australialaiset ammatinharjoittajat olivat huolissaan ohjelmisto-prosessien parantamisen tietoisuuden puutteesta, mutta he olivat enemmän huolissaan organisaatioiden käytännöistä ja tuen puutteesta.

Tutkimuksen mukaan vaikuttaisi siltä, että ammatinharjoittajat tiedostavat parhaiten ne ohjelmisto-prosessien parantamisen toteutumisen esteet, joihin ovat itse törmänneet ja jotka siten kokevat tärkeimmiksi. Vietnamilaiset ammatinharjoittajat löysivät yleisesti enemmän ohjelmisto-prosessien parantamisen toteutumisen esteitä kuin Australialaiset ammatinharjoittajat.

Johtopäätökset

Tutkimuksen perusteella riittävät resurssit, sponsorointi ja ohjelmisto-prosessien parantamisen tietoisuuden kouluttaminen auttaisi ohjelmisto-prosessien parantamisen toteutumista kehittyvissä maissa. Lisätutkimusta aiheesta myös suositellaan. Tutkimuksen toivotaan auttavan ymmärtämään niitä tekijöitä, jotka estävät ohjelmisto-prosessien parantamisen toteutumista, sekä auttamaan löytämään parempia tapoja toteuttaa ohjelmisto-prosessien parantamiseen tähtääviä toimintamalleja.

Jari Laitinen

Software development project success and failure from the supplier's perspective: A systematic literature review

P. Savolainen, J.J. Ahonen, and I. Richardson, International Journal of Project Management, in press, 2011

Johdanto

Lukemassani kirjallisuuskatsauksessa tarkasteltiin ohjelmistoprojektin onnistumista ohjelmistoprojektin toteuttajan näkökulmasta ja verrattiin sitä toimeksiantajan näkökulmaan. Onnistumista mitattiin asiakkaan tyytyväisyydellä, lyhyen ajan tuotolla ja pitkän ajan tuotolla. Kirjallisuuskatsauksen tarkoituksena oli koota yhteen viimeaikaisia tutkimustuloksia ohjelmistoprojektin onnistumiskriteereistä. Tutkittavia kriteereitä olivat sovitussa aikataulussa ja budjetissa pysyminen sekä projektin vaatimusten täyttäminen.

Kirjallisuuskatsauksen aineistosta löytyi eroavia mielipiteitä liittyen projektin laatua mittaaviin tekijöihin. Erään tutkimuksen mukaan tavanomaiset mittarit kuten budjetti, aikataulu ja vaatimusmäärittely muuttuvat useaan kertaan ohjelmistoprojektin aikana ja ovat siten virheellisiä mittareita. Vaikka vaatimusmäärittelyn muuttuminen kesken projektin onkin ohjelmistoprojekteille tyypillistä, todettiin puutteelliset vaatimukset projektin alussa ongelmaksi. Useimmat ohjelmistoprojektit aloitetaan ennen kuin aihealueen varsinainen problematiikka on täysin ymmärretty.

Tekstissä käsiteltiin erikseen sekä projektinhallinnan että projektin onnistuminen. Nämä eroteltiin siten, että projektinhallinta koostuu työvälaineiden ja menetelmien soveltamisesta kun taas projekti on kokonaisuus johon projektinhallinta osana kuuluu. Jako tapahtuu luontevasti projekteissa, joissa projektin toteutus on ulkoistettu. Projektinhallinnan laatua tulee tarkastella erikseen asiakkaan ja toteuttajan näkökulmista, sillä niissä saattaa olla eroavaisuuksia. Kärjistetynä asiakkaan tavoite on saada tarvitsemansa tuote mahdollisimman pienin kustannuksin kun taas projektin toteuttajan tavoite on maksimoida asiakkaan kustannukset ja samalla oma voittonsa.

Tulokset

Kirjallisuutta tarkasteltiin arvioimalla tekstin tapaa suhtautua projektin onnistumiseen toimittajan näkökulmasta. Lisäksi eroteltiin projektinhallinnan ja projektin toteuttajan näkökulmat. Kirjallisuutta rajattiin valistamalla vain tieteellisesti merkittäviä aikakausjulkaisuja. Tekstien haut toteutettiin järjestelmällisesti käyttäen loogisia lausekkeita ja samoja parametreja jokaisen tietokannan kohdalla. Löydetty artikkeli valittiin, mikäli siinä käsiteltiin jotain tämän tekstin johdanto-osuudessa mainittuja aiheita. Jokaisesta löydetystä artikkelista kerättiin tiedot taulukkoon. Taulukko sisälsi metatietoja kyseisestä tekstistä, kuten näkökulman, josta teksti on kirjoitettu, onnistumiskriteerit ja tekstin lähde.

Lähes kaikki tekstit käsitelivät projektinhallinnan ja projektin onnistumista ainakin toimittajan näkökulmasta. Noin puolet käsitelivät myös toimeksiantajan näkökulmaa.

Löydettyjä onnistumiskriteerejä olivat toimittajan voitot, asiakastyytyväisyys ja myöhemmät yhteistyömahdollisuudet. Lisäksi kiinnitettiin huomiota riskienhallintaan jonka pitäisi kattaa paitsi kyseisen projektin riskianalyysi, mutta myös liiketoiminnalle aiheutuvat riskit. Yksikään valituista artikkeleista ei käsitellyt epäonnistumisen kriteerejä.

Yhteenveto

Kirjallisuuskatsauksessa onnistuttiin määrittelemään yleisen käsityksen mukaiset ohjelmistoprojektin laatukriteerit projektin tarjoajan ja projektinhallinnan näkökulmasta. Yhteenvetona toimittajan näkökulmasta tärkeimmät asiat ovat asiakastyytyväisyys, lyhyen aikavälin tuotto ja pitkän aikavälin tuotto. Näin ollen liiketoiminnan jatkuvuutta pidetään yleisesti tärkeänä. Harvat mukana olleista tutkimuksista käsitelivät erikseen projekteja, joissa projektin toteutus on eriytetty projektinhallinnasta, jolloin näiden onnistumisia voitaisiin käsitellä erillisinä. Tällaisia projekteja tarkastelemalla tulokset saattaisivat erota tämän kirjallisuuskatsauksen havainnoista.

Antti Kantola

Project management by early warnings

I.O. Nikander and E. Eloranta, International Journal of Project Management, volume 19, issue 7, pages 385-399, 2001

Johdanto

Tässä artikkelissa annetaan yksi mahdollinen ratkaisu projektihallintaan. Projektijohtamisessa esiintyvät ongelmat on mahdollista välttää tarkastelemalla edellisten projektien tapahtumia ja näin ennustamalla tulevia riskejä tai uhkia. Riskienhallinta käsitteenä sisältää johtamisperiaatteiden ja käytänteiden järjestelmällistä hyväksikäyttöä riskien analysoimiseksi ja valvomiseksi. Riskianalyysi on selvitys, jolla pyritään löytämään mm. teknisistä laitteista, ihmisen toiminnasta sekä ympäristöolosuhteista riskejä synnyttäviä tekijöitä. Näistä aiheutuvia seurauksia arvioidaan sekä etsitään korvaavia ja edullisimpia toimenpidevaihtoehtoja. Riskianalyysillä tarkoitetaan prosessia, jossa jo olemassa olevaa tietoa kohdistetaan järjestelmällisesti uhkien tunnistamiseen sekä ihmisiin, omaisuuteen tai ympäristöön kohdistuvan riskin suuruuden arvioimiseen.

Tulokset

Projektinhallinta on nykypäivänä kiristetty äärimmilleen, sillä projektit pitäisi saattaa loppuun yhä aiemmin ja pienemmällä rahalla. (s. 385) Lisäksi asiakkaan painostus ja tiukalla aikataululla myyty projekti aiheuttavat lisää riskejä projektille. Projektipäällikön yksi tehtävistä on pitää kiinni aikatauluista sekä kustannuksista. Hänen pitäisi samaan aikaan pystyä näkemään ennusmerkkejä mahdollisista tulevista ongelmista. Ongelmat saattavat alussa olla pieniä, mutta aikanaan ne saattavat kaataa koko projektin. Vaativien ja suurien projektien vetämisessä tarvitsee olla kokenut projektipäällikkö, koska hänellä on kokemusta projektien vetämisen ”historiasta”. Katsomalla taaksepäin mikä projekteissa on yleisesti ollut riski tai ongelma, voidaan uudessa projektissa välttää tiedostamalla nämä heikkoudet. Kaikista ongelmista aikataululla on eniten riskitekijöitä. (s.386) Ongelmien todennäköisyys kasvaa nopeasti, kun ollaan valmistamassa tuotetta liian tiukassa aikataulussa. Kaikki tekijät, esimerkiksi vaikea tuote tai työntekijäpula, vaikuttavat suoraan aikatauluun ja näin ollen ne lasketaan osatekijöiksi aikatauluriskeihin.

Projektipäällikön rooli on elintärkeä projektin kannalta, koska hän on ainoa projektiryhmän jäsen, joka voi nähdä koko projektin ongelmat ja tulevat riskit. Jatkuvasti tarkkailemalla kustannuksia, aikaa sekä toteutusta, voidaan nähdä projektin ongelmakohdat varhain. Lisäksi projektipäällikön tulee olla yhteydessä projektiryhmäläisiin, haastatella heitä ja selvittää onko heillä edessään suuria ongelmia ja miten ne voitaisiin selvittää. Haastatteleamalla projektiryhmäläisiä, projektipäällikkö saa tärkeää tietoa projektin etenemisestä sekä pääsee lähemmäksi projektin toteutusta. (s. 389) Näin kokenut projektipäällikkö pystyy näkemään onko edessä mahdollisia ongelmatilanteita ja jo ennalta ratkaisemaan ne. Projektipäällikkö pystyy järjestelemään projektin toteutusta eli priorisoimaan, tietämällä varhaisessa vaiheessa riskeistä ja niiden laajuudesta. Projektipäällikön tehtävä on pitää huolta, että projekti etenee kohtuullisessa ajassa ja että riskit tai ongelmat on ainakin alustavasti mietitty eteenpäin.

Ennakkovaroituksiin perustuvassa projektinhallinnassa projektipäällikön rooli korostuu. (s. 397) Tällaisissa projekteissa johtajuus ja kokemus ovat avainasemassa, koska edellisten projektien kokemuksella voidaan ratkaista ennalta tulevat ongelmat. (s.398)

Johtopäätökset

Ennakkovaroituksiin perustuvassa projektinhallinnassa on tärkeää hallinnointi ja päätöksenteko. Väärillä päätöksillä voi olla katastrofaaliset seuraukset projektille. Projektipäälliköiden on pyrittävä tarkkailemaan projektia kokonaisuudessaan sekä sen jokaista vaihetta, tunnistaakseen mahdolliset tulevat tai piilevät ongelmat. Ongelmien varhaisella löytymisellä ehkäistään paniikki-tilanteiden ja yllätysten syntymistä.

Tero Strakh

The impact of knowledge coordination on virtual team performance over time

P. Kanawattanachai and Y. Yoo, MIS Quarterly, volume 31, number 4, pages 783-808, 2007

Johdanto

Nykyajan kiivaasti muuttuvassa liiketalousmaailmassa organisaatioille on entistä tärkeämpää kyky luoda ja välittää tietoa. Tämä on keskeistä kilpailukyvyyn ylläpitämiseksi. Tiiviit, mutta tilapäiset työyksiköt, tiimit, ovat tähän kykenevien organisaatioiden rakennuspalikoita. Tuoreet tutkimukset osoittavat, että kun tiimejä yhä enenevässä määrin muutetaan virtuaalisiksi, eli hajautetaan maantieteellisesti ja ajallisesti sähköisien yhteydenpitovälineiden avustuksella, muodostuu myös tiedon hallinnointi tiimin jäsenten kesken merkittäväksi haasteeksi.

Koska tätä etätyöhön liittyvää uudenlaista tiedonkulun haasteellisuutta ei ollut merkittävästi tutkittu, halusivat tutkimuksen tekijät lähteä selvittämään sitä, kuinka maantieteellisesti hajautetut etätyötiimit oppivat havaitsemaan kanssatiimiläistensä tietotaidon, luottamaan siihen ja koordinoimaan ammattitaitoaan tehokkaasti. He kehittivät tätä tutkimusta varten transaktiivisen muistin (transactive memory systems, TMS) keskeisten osien muuttumista ja tiimin tehokkuusvaikutusta ajan mittaan selventävän mallin. Nämä kolme transaktiiviseen muistiin liitettävää käyttäytymisulottuvuutta ovat: asiantuntemuksen sijainti (expertise location), tiedon koordinointi (knowledge coordination) ja havaintoihin pohjautuva luottamus (cognitive-based trust). Lyhyemmin näihin osiin viitataan sanoilla tunnistaminen (recognizing), luottaminen (trusting) ja koordinointi (coordinating).

Etätyön erilaisesta sosiaalisuudesta johtuen tutkimuksen tehtäväksi määräytyi lyhyesti se, voiko etätiimi, jolla ei ole kasvokkain tapahtuvia tapaamisia, muodostaa transaktiivisen muistin. Ennen kaikkea tutkijat yrittävät vahvistaa oletuksensa, jonka mukaan tiedon koordinointi olisi kolmesta mainitusta osasta merkittävin vaikuttaja.

Tulokset

Tutkimuksen aineistona käytettiin kolmea lähdettä: kyselyihin saatuja vastauksia, sähköisien viestien arkistoa, sekä objektiivista tiimien suorituskykytesteytystä (s. 789).

Keskeytysten jälkeen osanottajina oli 146 tällä tavoin kurssia suorittavaa MBA-ohjelman opiskelijaa kymmenestä erilaisesta kansallisuudesta, neljästä eri yliopistosta. Nämä osanottajat oli jaettu 38:een kolmi- ja nelihenkiseen tiimiin satunnaisesti siten, että vain kahdessa tiimissä oli edustajia samasta yliopistosta (s. 790). Kullekin tiimin jäsenelle annettiin satunnaisperiaatteella rooli tiimissä (s. 791).

Tiimien tehtävänä oli pelata kurssia varten suunniteltua ja etukäteen testattua Inc. 2000 -peliä. Peli painottaa tasaisesti neljää merkittävää liiketalousaluetta: markkinointia, taloutta, tuotantoa ja operaatioita, sekä henkilöitä. Kullekin tiimille annettiin hallittavaksi samanlainen yritys, tehtävänään peitota muiden tiimien yritykset (sivut 790-791).

Mittarina käytettiin yrityksen osakekurssia (s. 791).

Tässä tutkimuksessa tiimien tuli tehdä viikoittain päätökset koskien 25:tä muuttujaa. Kaikkien tiimien lähetettyä päätöksensä ylläpitäjä suoritti viikon tapahtumat. Pelin pituudeksi asetettiin kahdeksan viikkoa, joista kunkin suorituksen jälkeen tiimien saamat pisteet arvioitiin heidän tekemiensä päätöksien menestyksellisyyden perusteella (s. 791).

Johtuen siitä, että pelin kulkuun ja sitä kautta pisteetykseen vaikuttivat muidenkin tiimien päätökset, tiimien oli mahdoton arvata tehokkainta tapaa toimia. Tämä dynaamisuus kannusti tiimiläiset yhdistämään tilannetajunsa ja hyödyntämään toistensa osaamista parempien päätöksien aikaansaamiseksi (s. 791).

Webbipohjainen käyttöliittymä salli vain tietyn roolin haltijan syöttää tietyn yritysalueen muuttujia. Tällä pakotettiin keskinäiseen vuorovaikutukseen. Ainoat viralliset keskustelukanavat olivat tiimikohtaiset sovellukseen integroidut keskustelutietokannat ja sähköpostilistat. Sähköpostiviestit arkistoitiin seulontaa varten (s. 791).

Data-aineiston kyselyitä tehtiin kolme kappaletta: viikkojen 2, 5 ja 8 tulosten syöttämisen jälkeen, siten, ettei viikon päätöksien seurauksia julkaistu ennen kuin kyselyihin oli vastattu. Tiimien suorituskykyä mitattiin pelin laskeman tiimin yrityksen osakekurssin perusteella (s. 791). Sähköpostiviestit koodattiin sisältönsä perusteella kahteentoista luokkaan, jotka koostuvat kahdesta pääluokasta: tunnepitoisista ja tehtäväkeskeisistä (sivut 791-792). Lisäksi kyselyillä mitattiin transaktiivisen muistin osia: tunnistamista, luottamista ja koordinoitua. Näiden mittaukseen käytettyjen menetelmien epävarmuus huomioitiin tulosten analysoinnissa (sivut 792-793).

Yhteenveto

Tutkimuksen tulosten perusteella kestää muutaman viikon ennen kuin tiimiläiset kykenevät tunnistamaan, luottamaan ja koordinoimaan vertaistensa tietotaitoa. Tämän pariviikkoisen oppimisprosessin aikana tehtäväkeskeinen kommunikointi oli merkittävä suorituskyvyn vaikutin. Mitä aiemmin ja useammin tehtäväkeskeistä kommunikointia tapahtui, sitä tehokkaammin tiimille osoitettu projekti käynnistyi. Lisäksi alkuvaiheessa luodut käsitykset muista ja luottamus toisiin säilyivät elinvoimaisina läpi projektin elinkaaren. Vain negatiivinen tehokkuus projektin keskivaiheilla aiheutti jäsenistä muodostetun käsityksen uudelleenarviointia.

Kuitenkaan alussa muodostettu transaktiivinen muisti ei yksin kovan työnteon ja säännöllisen kommunikoinnin avulla edelleenkehittä tiimin tehokkuutta. Tutkimus osoittaa, että tehtäväkeskeisen kommunikoinnin suora merkitys katoaa ajanmittaen, tehtävätietouskoordinoinnin noustessa tärkeäksi suorituskyvyn tekijäksi. Johtopäätöksenä hyvin kehittyneen transaktiivisen muistin omaava tiimi voi kommunikoida "älykkäästi", minimoimalla tehtäväkeskeisen kommunikoinnin suorituskyvyn siitä heikentymättä. Tämän transaktiivisen muistin kehittyminen tälle tasolle voi kuitenkin kestää kauemmin kuin ei-etätiimeillä.

Koska koeympäristö oli tarkoin määritetty, tulee tulosten soveltamisessa olla tarkkana. Huomioonotettavia seikkoja ovat 1) tiimin jäsenten roolitus mielivaltaisesti tutkijoiden toimesta, kun todellisuudessa perusteena käytettäisiin todennäköisimmin osaamista; sekä se, että kukin jäsen oli vain yhdessä tiimissä, kun tyypillistä on se, että yksilö on

samanaikaisesti useassa 2) tehtävän sanelema tiimiläisten keskinäinen riippuvuus, jonka paljous sanelee transaktiivisen muistin hyödyllisyyttä 3) tiimiläisten valinta kourallisesta yliopistoja, joka teki mahdolliseksi sen, että jäsenet ovat voineet kommunikoida muiden tutkimukseen valittujen henkilöiden kanssa, vaikkakaan tätä selvittänyt analyysi ei paljastanut tätä tapahtuneen tai sen vaikuttaneen tiimien suorituksiin 4) tutkimuksen monimutkaisuudesta johtuen erilaisten kulttuurien ja aikaerojen edustajien merkitystä ei selvitetty, vaikka joissakin muissa tutkimuksissa on todettu näillä olevan negatiivinen vaikutus etätiimien koordinointiin ja kommunikointiin 5) tehokkuuden määrittäminen vain simulaation matemaattisesti määritellysti tuottaman tuloksen pohjalta, kun todellisuudessa muina mittareina voitaisiin perustellusti käyttää muun muassa tyytyväisyyttä tiimin prosessiin ja tuloksiin.

Joitakin yleishyödyllisiä havaintoja voidaan kuitenkin tehdä. Tarve keskinäistä vuorovaikutusta ja tiedonjakoa helpottaviin työkaluihin on eräs näistä. Esimerkkinä tällaisesta tarjotaan sähköinen arkisto, josta käsin selviäisi kunkin kanssatiimiläisen osaaminen.

Toisekseen etätiimiä kootessa on myös syytä painottaa varhaista ja toistuvaa tehtäväkeskeistä kommunikointia. Tällaiset jäänrikkojatapahtumat eivät kuitenkaan saisi viedä liiaksi huomiota muilta transaktiivisen muistin osilta. Lisäksi tehtäväkeskeisen kommunikoinnin painottaminen myöhemmässä vaiheessa voi johtaa tiimiä harhaan, kun hyödyllisempää olisi keskittyä tukemaan erityistietämyksen koordinoitua.

Vaikka tutkimus osoittaa, että kasvokkain tapaamaton tiimi voi muodostaa suorituskykyä tehostavan transaktiivisen muistin, vaikuttaisi siltä, että tämän muodostumiseen kuluu pidempi aika kuin laboratoriotesteissä on havaittu: jopa kahdeksan viikkoa. Tuloksista huolimatta tutkimuksen katsottiin kuitenkin synnyttäneen enemmän kysymyksiä kuin se antoi vastauksia, toimien vain askeleena kohti kattavamman tiedon antavaa tutkimusta.

Timo Virtanen

ABC-Sprints: Adapting Scrum to academic game development courses

J. Schild, R. Walter, and M. Masuch, in Proceedings of the Fifth International Conference on the Foundations of Digital Games, 2010

Johdanto

Pelituotannon opettaminen on haastavaa, koska pelien tekoon liittyy laaja kattaus erilaisia osakokonaisuuksia ja toteutusteknologioita. Tämä johtaa siihen, että kurssit joilla perehdytään alaan, ovat yleensä intensiivisiä tai kopiointipainotteisia, ja ne saattaa kestää useita lukukausia. Yhteen lukukauteen tähtäävissä kursseissa, aiemmin esitettyjen suositusten mukaan, tulisi lähestymistapa valita luennoinnin ja peliprojektin väliltä. Paperissa esitetään lähestymistapa, jolla luennointi ja projekti saadaan mahdutettua yhteen lukukauteen.

Ratkaisuksi molempien lähestymistapojen sovittamiseksi yhteen lukukauteen ei-intensiiviseksi kurssiksi, jossa osallistujan viikkotuntimäärän ei tulisi ylittää 6-8 tuntia, esitetään modifioitua Scrumia. Scrum-lähestymistapa on ollut pelialalla menestyksekkäs; sen avulla korkealaatuisia pelejä on saatu toteutettua niin, että aikataulut ovat pitäneet. Modifioituun Scrum-ohjelmistokehitysmalliin pohjaava kurssi on kestoaltaan 15 viikon mittainen ja se jakautuu neljään vaiheeseen: (i) Suunnitelma, viikot 2-5; (ii) Alpha sprint, viikot 6-9; (iii) Beta sprint, viikot 10-13; (iv) Completion sprint, 13-15. Ensimmäinen viikko menee yleisten asioiden hoitoon.

Suunnitelmavaiheessa luodaan ideoita, konsepteja, ryhmäydytään ja suunnitellaan. Jakson aikana luennoitiin aiheista: “Aiheet, innovaatio ja ideat” (vko 2), “Tuotantoprosessi” (vko 3), “Tuotantotyökalut” (vko 4), “Projektituotanto” (vko 5). **Alpha sprintin** tarkoituksena on tarjota “proof of concept”; onnistua luomaan pelin perusfunktionaalisuus, ja tämän lisäksi suunnitella beta sprintin backlog. Alpha sprintin aikana luennoitiin aiheista: “Pelin rakenne” (vko 6), “Gameplay ja balansointi” (vko 7), “Käyttöliittymäsuunnittelu” (vko 8). **Beta sprintin** tarkoituksena puolestaan on saada peliin kaikki se toiminnallisuus, joka sille on suunniteltu, ja suunnitellaan Completion sprintin backlog. Luento-osuuksilla käytiin läpi seuraava sisältö: Viikolla 10 järjestettiin joulupelitapahtuma, “Interaktiivinen tarinankerronta” (vko 11), “Hahmonkehitys” (vko 12). **Completion sprintin** aikana pyritään aikaansaamaan bugivapaa, viimeistelty, pelin lopullinen versio. Rinnalla pyörivä luento-osuus kattaa: “Koodin arviointi ja uudestaan rakenteellistaminen” (vko 13), “Peliliiketoiminta” (vko 14).

Merkittävä ero ABC-Sprints -lähestymistavan ja Scrumin välillä löytyy Scrum-masterin roolista, joka ABC-Sprintsissa on ryhmän sisäpuolella, siinä missä Scrumissa on ulkopuolella. Viikottainen työkuorma ABC-Sprintissa on maksimissaan 8 tuntia ja Scrum-tilaisuuksia järjestetään kaksi viikossa, päivittäisten tapaamisten sijaa. Kurssisuunnitelman mukainen kurssi käytiin läpi kolmella tiimillä, joissa lopulta oli yhteensä 13 henkeä. Toteutustyökaluna käytettiin XNA Game Studion versiota 3.1, josta oli aikaisempia hyviä kokemuksia.

Tulokset

Prosessin toimivuutta pyrittiin hahmottamaan kahdella tapaa; paperin julkaisijat kertovat kokemuksia vaihe vaiheelta ja ryhmätyöläisiltä kysyttiin anonyymisti mielipiteitä lähestymistavan eri osa-alueiden toimivuudesta.

Pelit valmistuivat ajallaan, toimivat sulavasti ja pitivät sisällään monia pelitasoja, jotka menivät reilusti konseptin demonstrointia pitemmälle. Verrattuna aikaisemmissa projekteissa kehitettyihin peleihin, pelit olivat monimutkaisempia, tarjosivat toiminnallisesti täysin eheän, viimeistellymmän ja tasapainotetumman pelikokemuksen.

Scrumiin suoraan liittyvissä kysymyksissä käytettiin skaalaa 1-5, from “very effective” to “not effective”. N=13, jos ei toisin mainita. “The students rated Scrum meetings as very effective ($\bar{x} = 1$; $\mu = 1.46$; $\sigma = .66$). Separating a project into Sprints was perceived as effective ($\bar{x} = 2$; $\mu = 1.85$; $\sigma = .80$), as was using a Sprint Backlog ($\bar{x} = 2$; $\mu = 2.54$; $\sigma = 1.13$). The Product Backlog ($\bar{x} = 3$; $\mu = 2.67$; $\sigma = 1.15$; N = 12;), the roles ($\bar{x} = 3$; $\mu = 2.57$; $\sigma = .62$; N = 12), and the Burndown Chart ($\bar{x} = 3$; $\mu = 3.08$; $\sigma = 1.5$) were regarded as neutral. The Burndown Chart caused the most inhomogeneous ratings with half of the students regarding it neutral and the others equally distributed to both sides of the scale.” (p. 191)

Arvioitaessa työmäärän jakautumista projektin ajalle, suunnitteluvaiheen kuormittavuus arvioitiin opiskelijoiden toimesta matalaksi, Alpha ja Beta -sprintit erittäin työläiksi, kuten myös Completion-sprintin, joka kuitenkin oli Alpha ja Beta -sprinttejä vähemmän kuormittava. Kokonaisuormittavuudeltaan projektit arvioitiin erittäin korkeaksi. Tuottavuus koettiin erittäin korkeaksi koko projektin ajalta, ja sen koettiin nousevan vaiheiden edetessä.

Opiskelijat arvioivat, että Scrumin käyttäminen olisi hyödyttänyt heitä muilla jo käydyillä kurseilla ja he suhtautuivat positiivisesti Scrumin kohtaamiseen jatkossa tulevilla kurseilla. 11/13 osallistujasta olisi valmis käyttämään Scrumia oma-aloitteisesti jatkossa.

Ryhmätyötä noin yleisesti koskien 12/13 koki, että ryhmätyö, yksilöllinen vastuu ja rakenteellinen suunnitelmallisuus ovat tärkeitä. Heistä yksilöllinen vastuu tiimeissä oli erittäin korkea ja projektityö vaikutti suhteellisen jäsenneytyltä.

Kirjoittajat kokivat saaneensa luentosisällön jaksotetuksi sopimaan pelikehitysvaiheisiin kohtuullisen hyvin, kuitenkin kaikkea tarpeellista sisältöä ei pystytty tarjoamaan aina ennen Sprintin alkua.

Yhteenveto

Kirjoittajat näkevät kurssisuunnitelman onnistuneen mainiosti, sen mahdollistaessa täysin pelattavien pelien aikaansaamisen suhteellisen lyhyessä ajassa, samalla harjoittaen opiskelijoiden ryhmätyötaitoja ja ajanhallintaa. Vaikka intensiivisyydestä ei lopulta päästyäkään eroon, he suosittelevat ABC-Sprints -lähestymistavan käyttöä muissakin pelikehityshankkeissa, oletuksella että opiskelijoita tuetaan Scrumin käyttämisessä.

Aleksi Tiensuu

Organizing self-organizing teams

R. Hoda, J. Noble, and S. Marshall, in Proceedings of the International Conference on Software Engineering (ICSE), 2010

Johdanto

Artikkelissa Organizing self-organizing teams [Hoda, Noble ja Marshall, 2010] pyritään vastaamaan kysymykseen siitä, miten ketterän ohjelmistokehityksen menetelmiä käyttävät työryhmät onnistuvat itseorganisoitumaan. Ryhmän itseorganisoitumisella tarkoitetaan tässä ryhmän sisäisten työtehtävien jakautumista ja tehtävien tukitoimintojen syntymistä ilman eksplisiittisesti määriteltyä, ryhmän ulkoista tai sisäistä johtamista.

Tutkimuksessa käytetty menetelmä, Grounded Theory (suom. ankkuroitu teoria), on järjestelmällinen tapa muodostaa teoriaa laadullisen tutkimusaineiston pohjalta. Tutkimusaineistosta etsitään toistuvia teemoja, jotka ryhmitellään tiiviimmiksi kuvauksiksi tasoittain:

Suora lainaus tutkimusaineistosta → Key Point → Code → Concept → Category (p. 287).

Menetelmällä tunnistettiin kuusi työryhmissä esiintynyttä, ryhmän kesken määrittelemätöntä roolia (Category), jotka helpottavat ryhmän itseorganisoitumista. Menetelmä on esitelty artikkelissa yleisellä tasolla ja tutkimuksesta esille nostetun esimerkkitapauksen (p. 287) avulla.

Tulokset

Aineisto, johon artikkelin tutkimus pohjautuu, hankittiin puolistrukturoiduin haastatteluin. Tutkimusta varten haastateltiin 24:ää henkilöä 14:sta yrityksestä, joiden koot vaihtelivat kymmenen ja 300 000:n työntekijän välillä. Haastatellut toimivat ketteriä Scrum- ja XP-ohjelmistokehitysmenetelmiä (eXtreme Programming) hyödyntäneissä projekteissa Intiassa ja Uudessa-Seelannissa, vaihtelevissa työtehtävissä: taloushallinnon johtotehtävät, liiketoiminta-analyysi, ketterien menetelmien valmennus (Agile Coach / Agile Trainer), tuotekehitys, testaus ja suunnittelu.

Grounded Theory-menetelmän mukaisesti haastattelut litteroitiin ja haastateltavien litteroiduista kommentteista etsittiin asiakohtaisia yhteneväisyyksiä. Löydetyt yhteneväisyydet jalostettiin suorista haastattelulainauksista avainlauseiksi (Key Point), jotka merkittiin koodein (Code). Koodit kuvaavat avainlauseet muutamalla sanalla, kuljettaen mukana viitettä alkuperäiseen lainaukseen asti.

Jokaista esiin nousutta asiaa verrattiin jatkuvasti jo olemassa olevaan koodattujen avainlauseiden kokoelmaan (Constant comparison, p. 287), jotta keskenään samat teemat tulisivat merkityiksi samalla koodilla.

Aiemmalla menettelyllä saadut koodit ryhmiteltiin käsitteiden (Concept) alle. Kategoriat (Category), eli tässä tapauksessa ketterän ohjelmistokehityksen menetelmiä käyttävien ryhmien henkilöroolit, saatiin liittämällä haastatteluista tunnistetut, haastateltujen mukaan samoissa henkilöissä esiintyneet käsitteet yhden kuvaavan roolinimikkeen alle. Rooleja

tunnistettiin kuusi: Mentor, Co-ordinator, Translator, Champion, Promoter ja Terminator.

Mentor (pp. 288-289) on ryhmän tukija ja ohjaaja, varsinkin työn alkuvaiheessa. Vaikka ketterät menetelmät eivät itsessään ole vaikeita oppia, niiden noudattaminen voi tuntua haastavan erilaiselta perinteisiin ohjelmistokehitysmalleihin tottuneille. Mentorin roolissa toimiva kasvattaa ryhmän yhteistä itsevarmuutta ketterien menetelmien käytössä ja kannustaa niihin kuuluvien tekniikoiden käyttöön jatkuvasti. Mentor myös korjaa ryhmässä esiintyviä väärinkäsityksiä ketteristä menetelmistä.

Co-ordinator (p. 289) toimii ryhmän edustajana. Vaikka ketterä ryhmä onkin itseään ohjaava, se tarvitsee edustajan, joka toimii asiakasrajapinnassa ja hoitaa keskustelun ja aikataulutuksen asiakkaan ja ryhmän välillä. Co-ordinatorin rooli on myös ottaa vastaan asiakkaalta tulevat vaatimusmuutokset ja auttaa ryhmää sopeutumaan niihin.

Translatorin (pp. 289-290) tavoite on parantaa asiakkaan ja ryhmän välistä ymmärrystä. Translator ymmärtää asiakkaan liiketoiminnalliset tavoitteet ja tulkitsee ne ryhmälle funktionaaliseksi ja teknisiksi vaatimuksiksi. Vastaavasti Translator ymmärtää työryhmän käyttämän teknisen kielen ja osaa selittää tekniset asiat tehokkaasti siten, että asiakas pystyy tekemään tietoisia liiketoiminnallisia päätöksiä selvityksen pohjalta.

Champion (pp. 290-291) puolustaa ketteriä menetelmiä oman organisaationsa ylimmälle johdolle selvittämällä, mitä menetelmiin kuuluu ja perustelemalla niiden käytön. Ketterät ryhmät eivät voi toimia eristyksessä omasta organisaatiostaan. Organisaatioissa on ymmärrettävä ketterien menetelmien hyödyt, jotta ryhmät voisivat noudattaa menetelmiä täysipainoisesti, joutumatta työtavoilleen haitallisten yleisten päätösten uhreiksi. Championin rooli on usein edellytys ketterien menetelmien käyttöönottoon organisaatioissa ylipäätään.

Promoterin (pp. 291-292) rooliin kuuluu asiakkaan vakuuttaminen ketterien menetelmien hyödyistä ja asiakkaan sitouttaminen projektiin. Monet ketterät menetelmät edellyttävät asiakkaan jatkuvaa sitoutumusta projektiin arviointi- ja vaatimusmäärittelytehtävissä.

Promoter on Championin tavoin ketterien menetelmien puolustaja, joka opettaa asiakkalle ketterien menetelmien toimintaperiaatteet ja niiden hyödyt. Tutkimusta varten tehdyistä haastatteluista huomattiin, että Promoterin ja Championin rooleissa toimi useimmissa tapauksissa sama henkilö.

Terminator (p. 292) tunnistaa ryhmän sisältä jäsenet, joiden toiminta estää ryhmää toimimasta tehokkaasti ja hankkii ylimmän johdon tuen heidän poistamiselleen.

Yksilöiden sopeutumattomuus ketteriin menetelmiin ja haluttomuus muuttaa toimintapojaan ovat suurimpia uhkia, joita ryhmä voi kohdata. Toisaalta myös joustamattomat ketterien menetelmien puhtaiden opinkappaleiden puolustajat voivat hidastaa ryhmän toimintaa. Joskus ryhmän sisällä voi vallita epätasapainoisuus, jota ryhmän jäsenet eivät osaa tunnistaa. Tällöin Terminator tunnistaa epätasapainon aiheuttavan tekijän ja tekee voitavansa poistaakseen sen.

Yhteenveto

Tutkimuksen tuloksena tunnistettiin kuusi itseorganisoituvassa työryhmässä esiintyvää, erikseen määräämätöntä henkilöroolia. Kaikki henkilöroolit löytyivät ainakin yhdestä

projektista sekä Intiassa että Uudessa-Seelannissa. On kuitenkin huomioitava, että tutkimuksessa käytetyn menetelmän, Grounded Theoryn, tuottamat tulokset ovat aina kontekstisidonnaisia.

Tutkimuksen tulosten voidaan ajatella olevan yleistettävissä muihin ketterän ohjelmistokehityksen menetelmiä käyttäviin työryhmiin, mutta ei kaikkeen ryhmätyöskentelyyn. Ketterien menetelmien parissa työskentelevät voivat tutkimuksen tulosten avulla ymmärtää ryhmien itseohjautuvuutta paremmin - sekä ryhmän jäsenen, että ketterien menetelmien valmentajan näkökulmasta.

Simo Pönni

Project management for the 21st century: Supporting collaborative design through risk analysis

J.L. Smith, S.A. Bohner, and D.S. McCrickard, in Proceedings of the 43rd Annual Southeast Regional Conference, pages 300–305, 2005

Johdanto

Ohjelmistojen koko kasvaa jatkuvasti ja monimutkaisuus lisääntyy, ja siksi järjestelmäsuunnittelun kokonaisuuden ymmärtäminen on haastavaa. Tieto on usein jakaantunut hajautetusti työskentelevälle ryhmälle (*distributed team*). Yhteistyö tällaisissa ryhmissä on vaikeampaa kuin perinteisesti samassa paikassa työskentelevien yhteistyö, joten hajautetuissa ryhmissä käytetään paljon tietokonesovelluksia, jotka mahdollistavat ryhmän kommunikoinnin ja yhteistyön.

Jotta työskentely olisi mahdollisimman tehokasta ja laadukasta, ryhmän pitäisi ylläpitää kokonaiskuva projektista. Ryhmän jäsenten pitäisi jakaa tietämystä ja myös tietää, mitä tietämystä he ovat jakaneet toisilleen. Kun tiimin jäsenillä näin on hyvä kokonaiskuva projektista, voidaan siinä keskittyä paremmin tärkeimpiin tavoitteisiin. Riskienhallinta on yksi tärkeä osa projektinhallintaa. Tiimin pitää keskustella mahdollisista riskeistä, sopia riskien tärkeysjärjestyksestä, jakaa riskien pienentämiseen liittyvä työ ja tarkkailla tilanteen kehittymistä koko projektin ajan.

Hajautetusti työskentelevä tiimi tarvitsee riskienhallintaprosessin käsittelyyn ja projektin kokonaisuuden ymmärtämiseen yhteistyötä tukevia (*collaborative*) sovelluksia, joiden avulla kommunikointi ja riskienhallinnan ylläpito onnistuvat. Tiimien käyttämien työkalujen tulisi tukea tehokasta yhteistyötä, eikä työkalujen käytöstä saisi aiheutua kohtuutonta ylimääräistä taakkaa työntekijöille. Projektinhallinnan työkaluissa tulisi siis olla toiminnallisuutta, jonka avulla tiimille saadaan muodostettua mahdollisimman hyvä tietoisuus tiimin toiminnasta (*activity awareness*), tehtävien jakamisesta ja riskienhallinnasta, jotka ovat oleellinen osa projektiin kuuluvaa suunnitteluprosessia.

Nykyiset yhteistoiminnalliset työkalut (*collaborative tools*) tukevat ryhmän jäsenten tietoisuutta toistensa toiminnasta esimerkiksi eri tavalla esitettyjen ilmoitusten (*notification*) avulla. Muun muassa SOPPTS, TeamSCOPE, TeamSpace ovat työkaluja, jotka on tehty helpottamaan tiimien työskentelyä. SOPPTS on tehtäväsuuntautunut projektinjohtamisen järjestelmä opiskelijaprojekteihin, TeamSCOPE:n avulla voidaan jakaa tiedostoja ja tarkastella jaettujen tiedostojen historiaa, ja TeamSpace järjestää palaverien tapahtumat aikajanelle. SoftRisk puolestaan on riskienhallinnan helpottamiseksi kehitetty järjestelmä, joka pyrkii automatisoimaan riskienhallintaprosessia.

Tulokset

Ohjelmistoalalla modulaarisen ohjelmoinnin ja olio-ohjelmoinnin myötä ohjelmistojen osista eli koodista on tullut uudelleenkäytettävää. Myös tuotekehitysprosessin ja

suunnitteluprosessien arkistoinnista ja uudelleenkäytöstä olisi hyötyä, niin että hyvät ideat muistettaisiin ja samat virheet tehtäisiin vain kerran. Suunnitteluprosessin aikana koetut asiat voisi tallentaa esimerkiksi opittuina asioina tai malleina, tai pienempinä yksiköinä, kuten yksittäisinä väitteinä (*claims*). Väitteistä kirjattaisiin myös hyvät ja huonot puolet, joista huonoja puolia voisi käyttää suoraan projektin riskianalyyseissä mahdollisina riskeinä. Huonoja puolia voi yhteensä olla tuhansia, joten ne täytyisi priorisoida, ja keskittyä projektissa vain pahimpiin riskeihin.

LINK-UP on joukko web-perustaisia työkaluja, joiden avulla väitteiden hallinta on mahdollista. LINK-UP -järjestelmä on yhteydessä väitteiden arkistoon, jolloin voidaan hyödyntää edellisten projektien kokemukset tehokkaasti. Järjestelmässä tehdään lista tuotteeseen liittyvistä väitteistä, ja ne järjestetään tärkeysjärjestykseen. Järjestelmää voitaisiin parantaa lisäämällä väitteiden listaan tuotesuunnittelun väitteiden lisäksi myös suunnitteluprosessiin liittyvät väitteet. Koko suunnitteluprosessin ajan tätä arkistoa myös laajennetaan uusilla väitteillä ja vanhoja väitteitä päivitetään. Jotta järjestelmä mahdollistaisi mahdollisimman tehokkaan työskentelyn projektissa, sen tulisi tukea hajautetusti työskentelevän tiimin yhteistyötä.

Projektinjohtamisessa tarvitaan systemaattista lähestymistapaa, ja tehokas riskienhallinta voisi olla parantamassa sekä suunnittelua että projektinjohtamista. Riskienhallintaan ja prosessien kokemusten uudelleenkäyttöön tarvitaan hyviä työkaluja, jotka mahdollistavat tämän kehityksen. Tehokas riskienhallintaohjelmisto auttaisi tiimin työntekijöitä keskittymään projektin tärkeimpiin asioihin. Siksi riskienhallintajärjestelmän tulisi jäsentää suunnitteluprosessi tärkeisiin, iteratiivisiin askeliin.

Järjestelmän tulisi auttaa tiimin yhteistyötä ylläpitämällä yhteistä, ulkoista yhteistä muistia. Tämän muistin tulisi sisältää sekä tuotteeseen että suunnitteluprosessiin liittyvät tiedot, kuten projektiryhmän jäsenten roolit, vastuut, osallistuminen ja edistyminen. Muistissa olevien tietojen tulisi olla helposti saatavilla kaikille ryhmän jäsenille.

Yhteenveto

Tehokkaan ja systemaattisen projektinjohtamisen mahdollistamiseksi ohjelmiston suunnitteluprosessissa tulisi noudattaa seuraavia periaatteita:

- Ohjaa suunnitteluprosessia riskienhallintamallin avulla.
- Tue tiimin yhteistyötä siten, että tiimi on tietoinen toiminnastaan.
- Esitä projektiin liittyvää tietoa aikapohjaisilla visualisoinneilla.
- Arkistoi tuotteeseen ja prosessiin liittyvä tieto niin, että sitä voidaan käyttää uudelleen.

Sopivat prosessit ja työkalut ovat avain menestyksekkääseen ohjelmistojen suunnitteluun. Lisäksi projektin aikana kertyneitä tietoja ja taitoja ei saisi heittää hukkaan projektin loputtua, vaan ne tulisi tallentaa uudelleenkäyttöä varten.

Projektin johtaminen ja tietojen uudelleenkäyttö mahdollistuvat paremmin, jos tiimin käyttämä järjestelmä sisältää myös toimivan riskienhallintamallin. Tuotteeseen ja tuotekehitysprosessiin liittyvän tiedon vertailu ja suunnitteluprosessiin liittyvien väitteiden määrittely ovat tärkeitä tehtäviä tulevaisuudessa.

Riskienhallintamallin käyttö, tiedon arkistointi ja uudelleenkäyttö sekä

suunnitteluprosessin arviointi ovat asioita, joiden avulla on mahdollista parantaa työryhmien suorituskkyä ja helpottaa projektin johtamista.

Minna Heinonen

Project management and national culture: A Dutch-French case study

J. de Bony, in Proceedings of International Conference on Organizational learning, Knowledge and Capabilities, 2009.

Johdanto

Yleinen käsitys on, että projektinhallintakäytäntöjä voidaan soveltaa yleismaailmallisesti. Kokemus työpaikoilla on kuitenkin usein osoittanut, että uudet, eri kulttuureista tulleet projektinhallintamenetelmät eivät olekaan soveltuneet sellaisenaan paikalliseen ympäristöön.

Uusia käytäntöjä sovellettaessa johto joutuu usein valitsemaan, pakotetaanko työntekijät seuraamaan uutta mallia, vai muokataanko menetelmiä paikallisen ympäristön ja kulttuurin mukaisiksi.

Kansallisen ympäristön vaikutusta projektinhallinnan toteuttamiseen tutkittiin tapaustutkimuksen kautta. Tutkimuksessa vertailtiin eroavaisuuksia ranskalaisen ja hollantilaisen kohderyhmän välillä, sekä keskityttiin erityisesti eroihin suhtautumisessa seurantaan ja arviointiin.

Ranskalais-hollantilaisessa yhteistyössä haluttiin yhdistää ranskalaisen ("Met") ja hollantilaisen ("Doel") kansallisen tutkimuskeskuksen voimavaroja tutkimusprojekteissa. Tapaustutkimuksen kohteena oli eräs tässä ympäristössä läpiviety projekti. Aiempien tutkimusten perusteella oli tiedossa, että kansallisen kulttuuritaustansa vuoksi hollantilaiset ovat tottuneet pitämään projektityöskentelyä normaalina työtapana. Ranskalaisten keskuudessa taas korostetaan yksilön itsemääräämisoikeutta työskentelyssä.

Tulokset

Tutkimus perustuu lähdekirjallisuuteen projektinhallinnasta, sekä kaikkeen saatavilla olevaan dokumentaatioon projektista. Lisäksi toteutettiin avoin haastattelu 10:lle projektiin osallistuneelle tutkijalle sekä 6:lle tutkimuskeskuksen johtotoportaaseen kuuluvalla. Myös muutamien haastattelijoiden kanssa keskusteltiin jälkepäin.

Doelin ehdotti, että projekti vietäisiin läpi sen omien prosessien mukaisesti, kuitenkin hieman muutettuna bilateraaliseseen ympäristöön soveltuvaksi. Vastuu projektista olisi molemmilla tutkimuslaitoksilla. Seuranta ja arviointi toteutettaisiin seuraavasti: kaksi asiantuntijaa kummastakin laitoksesta arvioisi vuosittain projektin edistymisen. Seuraavaksi syntyneet kaksi raporttia yhdistettäisiin ja raportti hyväksyttäisiin kummankin laitoksen hallitusten toimesta.

Todellisuudessa Doelin projektikoordinaattori kirjoitti aina vuosiraportit. Metin koordinaattori lisäsi loppuun vain muutaman rivin ja allekirjoituksensa. Metin johtoporras ei toteuttanut arviointia Doelin ehdotuksen mukaan ja vasta muutaman muistutuksen jälkeen arvioinnit kirjoitettiin, mutta paljon vapaammassa muodossa kuin

alunperin oli sovittu.

Doelin johtoportaan projektia pidettiin sopimuksena ja sopimuksen kohteena oli työsuunnitelma, jota piti kirjaimellisesti noudattaa. Puhuessaan projektista hollantilaiset käyttivät usein sanoja "yhteistyö", "työskennellä yhdessä" ja "yhteinen työ". He näkivät projektin tavoitteiden menevän yksilön tavoitteiden edelle - esimerkiksi tilanteessa, jossa henkilöt eivät olisi muuten tulleet hyvin toimeen keskenään.

Ranskalaiset eivät ottaneet tutkimuksen hallintaa projektina kirjaimellisesti. Haastattelijan mukaan ranskalaiset eivät arvostaneet projektia ja pitivät tällaista toimintamallia liian raskaana ja byrokraattisena. Doelilla talousjohtaja ei voinut ajatellakaan, että projektia rahoitettaisiin ilman mahdollisuutta seurata ja arvioida sitä. Metillä ei taas ollut totuttu tällaiseen käytäntöön, vaan haluttiin olla puuttumatta tutkijoiden työhön projektin aikana.

Metillä projektin arviointi ja seuranta aiheuttivat ristiriitaisia tunteita: toisaalta hyväksyttiin, että projektia on seurattava, toisaalta kyseenalaistettiin menetelmiä. Tutkijat kertoivat, että he osaavat työnsä ja ihmettelivät, miksi heitä piti tarkkailla.

Projektista puhuttaessa ranskalaiset käyttivät johtamisesta negatiivisia sanoja, kuten "naurettava", "idioottimainen" ja "karikatyyrinen". Ranskalaiset arvostelivat hollantilaisten tapaa laskea tunteja ja tehdä vain projektiin liittyviä asioita; he itse halusivat enemmän vapauksia.

Yhteenveto

Tämän tutkimuksen perusteella projektinhallinta koettiin ja toteutettiin eri tavalla ranskalaisten ja hollantilaisten keskuudessa. Koska Doel oli riippumaton markkinasuuntautunut laitos ja Met taas valtiojohtoinen instituutti, niin erot vielä kärjistyivät. Muutenkin hollantilaiset olivat enemmän projektisuuntautuneita; tämä selittyy osin konsensus-kulttuurilla: jo peruskoulussa oppilaita opetetaan hakemaan yhteisiä ratkaisuja ja toimimaan projektiluontoisesti.

Yleisesti, johtamiskäytännöt saadaan parhaiten toteutettua kun otetaan huomioon kansalliset piirteet ja muokataan käytäntöjä paikallisesti - mahdollisesti paikallisten johtajien avulla.

Pekka Mäkiäho

Information technology project management within public sector organizations

K.M. Rosacker ja R.E. Rosacker, Journal of Enterprise Information Management, volume 23, number 5, pages 587-594, 2010

Johdanto

Informaatioteknologiaprojektin (IT-projektin) hallinnasta on tutkimuskentällä kirjoitettu paljon (IT-projektilla tarkoitetaan projektia, johon voi sisältyä sekä laitteistoja että ohjelmistoja). Suurin osa tästä tutkimuksesta suuntautuu kuitenkin yksityiselle puolelle, vaikka informaatioteknologia (IT) on myös julkisella sektorilla erittäin tärkeässä asemassa. Julkisen sektorin informaatioteknologiaan käytetään lisäksi valtavia summia: Kirjoittajien lähteen (s. 587; Gross 2009) mukaan Yhdysvaltojen liittovaltio kulutti 76 miljardia dollaria IT:hen vuonna 2009. Julkisen sektorin organisaatiot ovat IT:n suurimpia hankkijoita, käyttäjiä ja kuluttajia. Lisäksi IT:n rooli julkishallinnossa on paitsi kasvava myös jatkuvasti uudelleenmuotoutuva suhteessa sidosryhmiin, Hyvänä esimerkkinä uudistumisesta ovat nykyään kansalaisille tarjottavat lukuisat e-palvelut ja sosiaalisen median ulottaminen niihin.

Julkisten organisaatioiden asema, sääntely, velvoitteet ja tarkoitus ovat erilaiset kuin yksityisen sektorin organisaatioiden. Monia julkisen sektorin IT-projekteja on kirjoittajien (s. 587) mukaan jälkeenpäin kuvattu ”tuhlaileviksi, tehottomiksi, huonosti hallituiksi, kalliiksi ja aikatauluistaan jääneiksi”. Projektinhallinnan parhaiden käytäntöjen tutkimisella ja tuntemisella voi olla merkittävä positiivinen vaikutus IT-investointien onnistumiseen. Koska yksityiset organisaatiot ovat luonteeltaan hyvin erilaisia kuin julkiset, ei niiden projektinhallinnasta hankittua tietoa voida siirtää julkiselle puolelle tutkimatta ensin niiden soveltuvuutta.

Tulokset

Kirjoittajat lähestyvät aihetta kolmella eri tavalla. Ensimmäiseksi he tekevät historiallisen katsauksen IT:n käyttöön julkisissa organisaatioissa. Toiseksi he tarkastelevat tietämystä projektinhallinnasta mahdollisimman laajana kokonaisuutena, ja kolmanneksi he etsivät ja tunnistavat julkisten organisaatioiden omaleimaiset piirteet. Lopuksi näiden konseptien avulla muodostetaan tutkimuksellinen viitekehys aiheesta. Artikkeliki keskittyy lähinnä Yhdysvaltojen julkiseen sektoriin tarkastellessaan julkisia organisaatioita.

Kirjoittajien mukaan IT oli julkisorganisaatioissa keskustietokoneiden aikaan lähinnä erillinen tukiyksikkö, joka avusti päivittäisissä rutiinitehtävissä ja joka oli johdon valvonnan ulkopuolella (s. 589). Eri toimipisteiden IT-asiantuntijat hallitsivat ja käsitelivät dataa ja laitteistoja. Vasta henkilökohtaiset työasemat aiheuttivat paitsi laskentatehon myös tiedon hallinnan hajauttamisen. Työasemista siirryttiin asiakaspalvelinarkkitehtuuriin ja viimein internetin aikakaudella moninaiseen ja hajautettuun IT:n hyväksikäyttöön. IT muuttui ”välttämättömästä pahasta” arvostetuksi johdon kumppaniksi, jolla voitiin paitsi tyydyttää tietotarpeita myös tehostaa toimintaa ja leikata

kustannuksia. Informaatioteknologiaan panostettiin, ja sen mahdollisuuksia haluttiin ymmärtää paremmin myös organisaation ydinstrategian kannalta.

Nykyisen ymmärryksen valossa organisaatioiden tulee kirjoittajien lähteen (Schwalbe, 2010) mukaan olla perehtynyt uusiin projektinhallinnan tekniikoihin (s. 589). Toisen lähteen (Jurison, 1999) mukaan kaikki IT:n toteutushankkeet käyttävät jonkinlaista projektinhallintaa. Näistä syistä ja suuren mittaluokan haasteista, ongelmista ja vaikutusalueesta huolimatta parhaiden käytäntöjen ja tekniikoiden soveltamisesta julkisen sektorin IT-projektinhallintaan on vähän empiiristä tutkimusta. Tämä on kirjoittajien mukaan suuri puute (s. 590).

Vaikka artikkelissa pyritään painottamaan julkisten organisaatioiden erilaisuutta suhteessa yksityisiin organisaatioihin, kirjoittajat huomioivat myös niiden samankaltaisuuden (s. 590): Molemmat ovat tilintarkastusvelvollisia ja molempien on kohdennettava niukkoja resursseja, noudatettava lakeja ja säädöksiä sekä ylipäättään toimittava järkevästi ja tehokkaasti. Organisaatiokulttuurien merkittävimmät erot ovat kirjoittajien mukaan julkisen sektorin kilpailijoiden puuttuminen (yleensä) sekä merkittävästi laajemmat sidosryhmät; julkisorganisaation projektijohtajat ovat vastuussa usein paitsi suoraan asiakkaalle, myös poliittiselle johdolle ja joukolle eri viranomaisia, työntekijöille, intressiryhmille sekä medialle. Monopoliasema ei pakota innovatiivisiin IT- investointeihin ja kausipolitikointi rikkoo pitkän tähtäimen suunnitelmia. Julkisorganisaatiot eivät myöskään voi pitää strategisia päätöksiään salaisina, mikä altistaa strategialinjaukset eri intressiryhmien painostukselle hyvin aikaisessa vaiheessa.

Vaikka laki säätelee myös yksityisiä organisaatioita, julkisorganisaatiot ovat kirjoittajien mukaan usein alisteisia paljon tarkemmalle sääntelylle (s. 590). Esimerkiksi monet projekteihin ja hankintoihin liittyvät menettelyt voivat olla lakisääteisiä. Vaikka ponnin seurata ”parhaita käytäntöjä” onkin hyvä, empiiriseen tutkimukseen pohjautuva tapauskohtainen menetelmien soveltaminen olisi kirjoittajien mielestä parempi lähestymistapa (p. 591). Taustalla vaikuttaa 1980-luvulla alkanut managerialistinen suuntaus NPM (New Public Management), jossa yritysmaailman käytänteitä alettiin voimakkaammin soveltaa julkisella sektorilla. Kirjoittajien mukaan viime vuosikymmenenä julkisorganisaatioiden IT:ssä yleistyneet ERP- ja CRM-projektit ja E-hankinta ovat esimerkkejä managerialistisesta hallintatavasta. Näiden projektien suuri epäonnistumissuhde on kuitenkin esimerkki menetelmien adoptoinnin ongelmallisuudesta. Empiirisen tutkimuksen osuus on sektoreiden erilaisuus huomioiden varsin rajoittunutta: Empiiristä tutkimusta on paljon IT-projektinhallinnasta ja julkisen organisaation hallinnasta (*management*), mutta ei IT-projektinhallinnasta julkisessa organisaatiossa (s. 591).

Yhteenveto ja pohdinta

Teknologia vaikuttaa kasvavassa määrin kaikkeen julkisen sektorin toimintaan, mutta IT-projekteihin johtajat ovat kuitenkin huonosti valmistautuneita ja huonosti varustautuneita (s. 591; Melin ja Axelsson 2009). Kun tyypillisesti julkisen sektorin projekteissa on lisäksi merkittävä pääoma, suuret henkilöstö- ja informaatioresurssit ja poliittisia sitoumuksia resurssien ja tietämyksen epäsuhta on huomattava (s. 592; Sharifi ja Manian 2010). Artikkelissa painotetaan empiirisen tutkimuksen tärkeyttä tämän epäsuhtan korjaamiseksi sekä ”oikeiden” parhaiden käytäntöjen löytämiseksi julkisen sektorin IT-

projektihallintaan.

Artikkeli oli mielestäni erittäin mielenkiintoinen. Vaikka näkökulma oli amerikkalainen, mielestäni artikkeli oli riittävän yleisluontoinen yhtymäkohtien löytymiseksi: Minusta kaikki päähuomiot pätevät myös Suomessa. Kirjoittajien luettelemat julkisen sektorin IT-projektien ongelmathan ovat sinänsä samanlaisia kuin kirjallisuudessa esiintyvät yksityisen sektorin ongelmatkin: budjetin ja toteutusaikataulun ylittyminen sekä sidosryhmien tyytymättömyys toiminnallisten vaatimusten toteutukseen (s. 593). Ongelmien samanlaisuudesta ei kuitenkaan välttämättä seuraa se, että ratkaisumallit olisivat samanlaisia. Tämä on mielestäni todella merkittävä havainto.

Pidän melko yllättävänä, että aiheesta on vain vähän empiiristä tutkimusta; julkisorganisaatioiden IT-ongelmat tuntuvat olevan mediassakin esillä melko taajaan, ja esimerkiksi terveydenhuollon tietojärjestelmien ongelmista on keskusteltu Suomessa kohta 30 vuotta. Lisäksi julkiset organisaatiot ovat, kuten artikkelissakin mainitaan, suunnattoman suuria rahankäyttäjiä ohjelmisto- ja IT-hankinnoissa. Tutkimusaiheen voisi kuvitella kiinnostavan sekä tekijöitä että tilaajia, mutta ei välttämättä aiheen suhteen hyvässä asemassa olevia ohjelmistotoimittajia.

Juhani Linna

An integrated approach to project management through classic CS III and video game development

U. Wolz and S.M. Pulimood, in proceedings of the 38th SIGCSE technical symposium on computer Science education, pages 322-326, 2007

Johdanto

Ohjelmistokehitysprojektien onnistumisprosentti on historiallisessa tarkastelussa ollut varsin pieni. Tutkimusten mukaan tämä on yleensä seurausta huonosta projektin johtamisesta. Projekteissa hyvin menestyvien yritysten taustalla on usein projektinjohtamiseen liittyviä periaatteita, joiden mukaan myös yritysten projektipäälliköt ovat saaneet koulutusta. Lisäksi nykyaikainen ohjelmistokehitys on kehittynyt suuntaan, jossa tietojenkäsittelijöiden tulee oppia kommunikoidaan muiden alojen työntekijöiden kanssa, sillä tietojenkäsittelijät harvoin eristetään täysin esimerkiksi asiakkaista. Teknisen johtajan kanssa kommunikointi ei siis enää riitä.

Opiskelijoille opetettiin kommunikointia ja projektinjohtamista ensin eräänlaisella peruskurssilla, jossa keskityttiin lähinnä teoriaosuuteen ja pienryhmäharjoituksiin. Tämän jälkeen opittuja asioita sovellettiin käytännössä toisella kurssilla, jossa luotiin laajan ryhmän (noin 20 henkeä) voimin videopeli. Ryhmään kuului opiskelijoita useilta aloilta, jolloin tietojenkäsittelyn opiskelijat pääsivät opettelemaan kommunikointia muiden alojen asiantuntijoiden kanssa.

Lisäksi artikkelissa tuodaan ilmi, että tietojenkäsittelystä kiinnostuneiden opiskelijoiden lukumäärä on ollut laskussa, ja että tietojenkäsittely nähdään usein raskaana, pelkästään ohjelmointia sisältävänä alana. Ryhmäpainotteisen kurssin avulla pyrittiin siis myös lisäämään muiden alojen opiskelijoiden kiinnostusta tietojenkäsittelyyn ja tietoisuutta siitä, mitä kaikkea ala pitää sisällään.

Tulokset

Tulokset saatiin tarkkailemalla opiskelijoita kahden kurssin aikana sekä haastatteleamalla heitä niiden jälkeen. Myös tenttimenestys huomioitiin, kun siihen suhteutettiin opiskelijoiden aikaisempi tietotaito ja heidän osallistumisensa toiselle tai molemmille esitellyistä kursseista.

Ensimmäisellä kurssilla opiskelijoille opetettiin muun muassa C++-ohjelmointia, ohjelmistokehitysprosessien perusteita, sekä pyrittiin kehittämään suullista ja kirjoitettua kommunikointia. Kurssin painopistettä siirrettiin enemmän viimeksimainittuun, ja sen lisäksi käytiin läpi esimerkiksi ryhmässä toimimista ja projektinjohtamiseen liittyviä ongelmia. Opiskelijat suorittivat kurssin aikana erilaisia tehtäviä, ja sen lisäksi opiskelijat jaettiin 4-5 hengen tiimeihin, joissa työskenneltiin koko kurssin läpi. Tiimit saivat päättää melko vapaasti oman rakenteensa sekä kommunikointivälineensä. Noin kurssin puolessavälissä jokaisesta tiimistä arvottiin yksi henkilö, joka uudelleensijoitettiin toiseen tiimiin. Tämä valmensi opiskelijoita tosielämän skenaarioita varten, sillä työelämässä

ihmiset esimerkiksi sairastuvat tai vaihtavat työpaikkaa jatkuvasti.

Opiskelijat itse vahvistavat, että kurssista on ollut hyötyä tulevia projektitöitä varten, ja useimmat kertovat hyötynensä siitä myöhemmin myös työelämässä ja työnhaussa. Kurssi myös valmisti opiskelijoita seuraavaan, laajempaa ja useita aloja yhdistelevää kurssia varten.

Toisella kurssilla käsiteltiin videopelejä. Tätä pidettiin hyvänä lähtökohtana, sillä tekniset taidot eivät yksinään riittäneet, vaan opiskelijoiden piti keskittyä myös esimerkiksi käsikirjoitukseen, äänimaailmaan, grafiikkaan ja pelattavuuteen. Videopeliprojekti oli opiskelijakeskeisellä kurssilla hyvä siksikin, että opiskelijat tuntevat projektin helpommin ”omakseen”. Kurssi kesti kaksi lukukautta, mutta oli jaettu suurpiirteisesti kahteen osaan. Ensimmäisessä osassa keskityttiin pelin suunnitteluun, ja toisessa toteutettiin suunnitellusta pelistä pelattava demo.

Monimutkainen, monihenkkinen ja useiden alojen tietotaitoa vaativa projekti vaatii luonnollisesti myös hyvää projektin organisointia. Opiskelijoiden piti huolehtia, että välitavoitteet saavutettiin ja projekti eteni aikataulussa. Eri henkilöiden vastuualueiden määrittäminen oli luonnollisesti myös tarpeellista.

Tekstissä todetaan, että otanta on toistaiseksi liian pieni yleispätevien johtopäätösten tekemiseksi, mutta että tulokset antavat joka tapauksessa viitteitä siitä, että toteutettu kahden kurssin käytäntö toimii opetustarkoituksiin varsin hyvin. Lisäksi toisen kurssin lopussa pidettiin tentti, jossa käsiteltiin projektinjohtamiseen ja kommunikointiin liittyviä kysymyksiä. Tentissä menestyivät parhaiten ne, jotka olivat mukana molemmilla kursseilla alusta loppuun.

Yhteenveto

Opiskelijat näyttävät tarvitsevan sekä teoreettista että soveltavaa ja haastavaa opetusta toimiakseen työelämässä tehokkaasti projekteissa, jotka yhdistelevät eri alojen asiantuntijoita. Tämä on olennaista, sillä nykyaikaisessa ohjelmistokehityksessä ja yleensäkin tietojenkäsittelijöitä tarvitsevissa projekteissa tietojenkäsittelijät eivät ole erillään muista, vaan heidän on kyettävä kommunikoimaan kaikkien toimijoiden kanssa niin sanotusti ”yhteisellä kielellä”.

Opiskelijat itse tunnistavat tarpeen opiskella ryhmässä työskentelyä ja projektin organisointia. Heidän mielestään kursseista on hyötyä myös työnhaussa ja työelämässä.

Ville Mäkelä

Deliberate ignorance in project risk management

E. Kutsch and M. Hall, International Journal of Project Management, volume 28, issue 3, pages 245–255, 2010

Johdanto

Project Management Institute (PMI) ja Association of Project Management (APM) antavat menetelmäsuosituksia projektinhallintaan. Näiden instituutioiden suosituksissa riskienhallinta on yksi tärkeimpiä projektinhallinnan osakohteita. Suositukset väittävät että projektin riskienhallinnan avulla projektipäälliköt voivat menestyksekkäästi hallita riskeihin liittyvää tietoa.

Panostukseen (input) keskittyvää tutkimusta ei juuri ole ollut, vaan on keskitytty tulosten tutkimiseen. Tämä tutkimus paikkaa aukkoa ja käsittelee kuinka ”*merkityksettömyys*” (tarkoituksellinen riskiin liittyvän tiedon huomiotta jättäminen) näkyy projektienhallinnassa ja kuinka se vaikuttaa johtamisen tehokkuuteen.

Riskienhallinnasta yleensä ajatellaan, että ensinnäkin sillä määritetään ongelmat ja toiseksi yksilöidään niistä ne, jotka voivat vaikeuttaa päämäärien saavuttamista. Kolmanneksi analyysillä määritetään riskien seuraukset ja todennäköisyys. Neljänneksi suunnitellaan riskeihin vaikuttamista: miten vanhoja riskejä seurataan, uusia tunnistetaan ja miten riskejä voidaan välttää ja kuinka niistä voidaan toipua. Pyrkimyksenä on saavuttaa suurin varmuus ja tietämättömyyttä, tiedon puutetta, pyritään välttämään.

Tässä tutkimuksessa päinvastoin määritellään tarkoituksellinen välinpitämättömyys *merkityksettömyydeksi (irrelevance)*: tietty informaatio jätetään tarkoituksella käsittelemättä.

Tulokset

Tutkimuksen metodina oli haastattelu. Erikokoisiin ja -pituisiin projekteihin osallistuneilta 18 projektipäälliköltä kysyttiin, kuinka he järjestivät riskienhallinnan ja millaisia ongelmia he kohtasivat. Sitten haastateltiin lisäkysymyksiä heitä, joiden toiminnassa *merkityksettömyys* näkyi. Haastatteluissa ei käytetty tarkkoja kysymyksiä vaan keskusteltiin aiheen mukaan. Löydöksistä odotettiin aiemmassa tutkimuksessa havaittuja kolmea *merkityksettömyyden* osa-aluetta: aiherajaukseen, päättämättömyyteen ja tabuihin liittyviä riskejä. Lisäksi tarkasteltiin, löytyisikö muita osa-alueita.

Ensimmäiseksi havaittiin tyyppi ”aiheiden rajaus” (*untopicality*). Projektipäälliköt keskittyivät teknisiin ongelmiin mutta pitivät muita merkityksettöminä. Näytti siltä, että päälliköt keskittyisivät helppoon riskienhallintaan eikä niinkään hyödylliseen riskienhallintaan. Keskityttiin ongelmiin, jotka olivat melko hyvin määriteltyjä tai joiden ratkaisuja voitiin kokeilla tai hylätä, tai jotka olivat samantapaisesti ratkaistavissa olevia samankaltaisia ongelmia.

Ratkaisemattomilla riskeillä (*undecidability*) tarkoitetaan sellaisia, joista ei päästä yhteisymmärrykseen. Tällaisiin saattoi johtaa muun muassa projektiin osallisten piiloagendat, joka vaikeutti luottamuksen rakentamista projektin osallisten kesken. Tämä

johti siihen, että riskienhallinta keskittyi selvempiin ja käsiteltäviin riskeihin.

Tabu (*taboo*) on sellainen riski, josta ei uskalleta puhua. Haastatteluissa kävi esimerkiksi ilmi, että eräät projektipäälliköt eivät halunneet kertoa osakkaille tietyistä riskeistä, koska se olisi saattanut johtaa projektin enneaikaiseen lopettamiseen.

Suspension of belief -riskit jätetään huomiotta, koska niiden ei uskota realisoituvan. Riski pyritään hoitamaan vain jos se on realisoitumassa. Kun ei etukäteen mietitä ”kuvitteellisten” riskien hallintaa vaan keskitytään ”todellisiin” riskeihin, säästetään aikaa ja rahaa.

Lopuksi arvioitiin, miten nämä neljä tyyppiä vaikuttivat projektien tehokkuuteen. Ratkaisemattomuusongelmat liittyivät todennäköisyyksien analysointiin, tabut vaikuttivat riskien suunnitteluun ja yksilöintiin, aiheiden rajauksesta tuli ongelma, kun määriteltiin vastuunkantajaa. Riskinhallinnan siirtäminen myöhemmäksi (*suspension*) liittyi lähinnä aikaan ja rahan ja siihen, käytetäänkö varsinaista riskienhallintaa lainkaan. Kaiken kaikkiaan projektipäälliköiden mahdollisuudet vaikuttaa riskeihin pienenevät näiden toimien seurauksena.

Tulosten pohjalta tekijät kehittivät mallin, jolla riskin merkityksellisyys voidaan selvittää. Ensiksi saavutetaan yhteisymmärrys riskistä projektiin osallisten kanssa (estetään aiheiden rajausta), sitten tarkistetaan, onko riski todellinen vai kuviteltu (ratkaisemattomuus vähenee) ja lopuksi selvitetään rahan- ja ajankäytön puitteissa (*suspension of belief*) riski-informaation hyödyllisyyttä (pyritään estämään tabujen syntymistä). Jos riski läpäisee kaikki seulat, on se merkityksellinen.

Johtopäätökset

PMI tai APM väittävät, että riskien yksilöimisen, analyysin ja niihin vastaamisen avulla projektipäälliköt saavuttavat suunnitellut tavoitteet. Tutkimusta ei juuri ole menetelmän todellisesta tehokkuudesta.

Tämän tutkielman tulokset osoittavat, että joissakin projekteissa riskienhallinta hoidetaan sivuuttamalla riskejä tietoisesti. Neljä *merkityksettömyyden* lajia näyttävät alentavan riskienhallinnan tasolle, jolla ei näytä olevan lainkaan tai on vain vähän vaikutusta projektin tuloksiin. Näyttää jopa siltä, että jos merkityksettömyyteen ei kiinnitetä huomiota, riskienhallinnasta voi tulla tehotonta tai se voi jopa vaikuttaa kielteisesti.

Käsitellyt projektit olivat IT-alalta eikä ole sanottua, että tuloksia voisi yleistää vaikkapa rakennusprojekteihin. Projektin erityispiirteet (koko, monimutkaisuus) eivät ole rajoittava tekijä mutta sosiaaliset olosuhteet voivat olla (kuten organisaation kulttuuri tai kansallinen kulttuuri).

Mika Kähkönen

Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation

M. Laanti, O. Salo, and P. Abrahamsson, Information and Software Technology, volume 53, issue 3, pages 276-290, 2011

Johdanto

Huolimatta sekä tieteellisillä että ei-tieteellisillä foorumeilla huomioituista ketteriin menetelmiin siirtymisen eduista, puuttuu laajempaa tutkimustietoa asiasta. Erityisesti kvantitatiivista näyttöä on vähän; samoin tietoa ketterien menetelmien soveltuvuudesta suuriin organisaatioihin. Lisäksi vähäininkin tutkimustieto on painottunutta. Viimeisimmät tutkimukset ovat suuntautuneet Extreme Programming -menetelmään.

Nokiolla tapahtuneen ketteriin menetelmiin siirtymisen yhteydessä, vuonna 2008, tehdyn tutkimuksen avulla pyritään tarjoamaan empiiristä todistusaineistoa skaalautuvasta Scrum-menetelmästä. Tutkimuksessa selvitetään, onko vastaajien asenteilla ketteriin menetelmiin yhteyttä kokemukseen ketteristä ja perinteisistä menetelmistä. Kokemusta mitataan sillä, kuinka pitkään on käyttänyt kyseistä lähestymistapaa. Lisäksi tutkimus tarjoaa tietoa ketterien menetelmien haasteellisimmista ja hyödyllisimmistä osa-alueista.

Teollisuuden näkökulmasta tulokset osoittavat, että harjaantuneet ketterien menetelmien käyttäjät huomaavat menetelmässä lisäarvoa. Menetelmän omaksuminen koettiin kaikkein haasteellisimmaksi. Eniten hyötyä koettiin saatavan vaatimustenhallinnassa ja läpinäkyvyydessä; tosin vaatimustenhallintaa pidettiin myös haasteellisimpina läpi koko tutkimuspopulaation.

Tulokset

Ensimmäinen tutkimushypoteesi oli: "Jos ketterät menetelmät eivät tarjoaisi todellista edistystä, menetelmän omaksijat olisivat aluksi innokkaita mutta kääntyisivät pessimismistiseksi saatuaan käytännön kokemusta ketteristä menetelmistä (s. 278)." Toinen oletus oli, että aikaisempi kokemus perinteisistä menetelmistä saattaisi vaikuttaa asenteisiin ja mielipiteisiin kielteisesti. Jälkimmäistä tutkittiin kvantitatiivisesti mitatuilla tutkimuskysymyksillä: "1) Kuinka ammatinharjoittajien kokemuksen pituus ketteristä menetelmistä vaikuttaa heidän suhtautumiseensa ketteriin menetelmiin? 2) Kuinka ammatinharjoittajien kokemuksen pituus perinteisistä menetelmistä vaikuttaa heidän suhtautumiseensa ketteriin menetelmiin? (s. 278)"

Tutkimusotos koostui yli tuhannesta vastaajasta seitsemässä eri maassa Euroopassa, Pohjois-Amerikassa ja Aasiassa. 90 % vastaajista oli tutkimuksesta ja kehityksestä. Myös osuus, joka tunnisti roolinsa ketterässä kehityksessä, oli 90 %. Vastaajien kokemus perinteisistä menetelmistä luokiteltiin viiteen luokkaan ja eniten vastaajia, 33 %, sijoittui 5 - 10 vuoden kokemukseen. Ketterien menetelmien kokemukset luokiteltiin kuuteen luokkaan, joista "ei käyttökokemusta" -luokkaan asettui reilu 20 % vastaajista. Eniten vastaajia oli luokassa 7-11 kuukauden käyttökokemus, 28 %.

Tutkimusaineisto kerättiin web-pohjaisella kyselylomakkeella kahden viikon aikana

huhtikuussa 2008. Vastaamisprosentti oli 33.

Kahdeksan väittämää yhdeksästä ketterien menetelmien vaikutuksista oli arviointiasteikolla myönteisemmällä puolella ainoastaan yhden väittämän, “ketterä kehitys tekee työstä vähemmän hektistä” jäädessä kielteiselle puolelle. Myönteisen suhtautumisen saaneisiin seikkoihin kuuluvat mm. kasvattaa kehittämisen tehokkuutta, tuotteen laatua, kehittämisen avoimuutta ja yhteistyötä, tekee työstä hausempaa ja suunnitelmallisempaa, kasvattaa kehitysryhmien autonomisuutta sekä mahdollistaa varhaisemman virheentunnistamisen.

Kysymykseen haluamisesta palata vanhaan työskentelytapaan vastasi kielteisesti 60 % yhdeksän prosentin halutessa palata entiseen. Neutraalin vastauksen (31 %) jättämistä selittää eniten ketterien menetelmien käyttökokemuksen puuttuminen kokonaan.

Varianssi- ja klusterianalyysin avulla tutkittiin sekä perinteisten menetelmien että ketterien menetelmien käyttökokemuksien pituuksien ja ketteriin menetelmiin suhtautumisen yhteyttä. “Voidaan päätellä, että pitkäaikainen kokemus ei-ketteristä menetelmistä vaikuttaa negatiivisesti tiettyihin mielipiteisiin (kasvava tehokkuus, työn suunnitelmallisuus ja varhainen virheiden löytäminen) ketterästä kehityksestä (s. 283).”

Laadullisin analyysimenetelmin selvitettiin avoimen kysymysten vastauksia ketterien menetelmien haasteista ja eduista, jolloin haasteiksi nousi erityisesti kolme kohtaa: käyttöönotto, vaatimusten hallinta / iteratiivinen suunnittelu sekä näkyvyys ja avoimuus.

Yhteenveto

Kun on tutkittu ammatinharjoittajien suhtautumista ketteriin menetelmiin suhteessa ammatillisen kokemuksen pituuteen voidaan väittää, että pidempi kokemus ketteristä menetelmien käytöstä johtaa myönteiseen kantaan. Ketteriä menetelmiä käyttäneiden joukossa tämä myönteisyys muuttui sitä suuremmaksi, mitä enemmän käyttökokemusta oli takana. Vain ainoastaan perinteisiä menetelmiä käyttäneiden joukossa esiintyi kielteistä suhtautumista, kun kokemusta oli yli kolme vuotta. Kuitenkin tutkimuskysymyksen “Haluaisitko palata vanhaan?” tulos voittaa em. kielteiset ennakkoluulot. Teollisuuden näkökulmasta nämä tulokset osoittavat, että kokeneet ketterien menetelmien käyttäjät voivat löytää lisäarvoa menetelmästä.

Toinen tutkimustavoite, ketterien menetelmien hyötyjen ja haasteiden löytäminen, nosti esiin kolme haastavaa aluetta: siirtyminen menetelmän käyttämiseen, vaatimusten hallinta ja suunnitteleminen joustavasti ja iteratiivisesti sekä avoimuus ja näkyvyys.

Tutkimus tehtiin yhdessä yrityksessä, jossa ketteriin menetelmiin siirtyminen oli tehty tietyllä tavalla. Tämän vuoksi tulokset eivät välttämättä ole sovellettavissa yleisesti. Lisätutkimusaiheina voisivat olla organisaation kulttuurin ja vaihtoehtoisten lähestymistapojen vaikutusten selvittäminen. Olisi myös kiintoisaa uusia tutkimus ketterien menetelmien käytön ja tuntemisen lisääntyä. Tutkimuksen tuloksia voi vääristää se, että ketteriä menetelmiä ovat alkaneet käyttää ensimmäisenä ne, jotka ovat olleet siitä kiinnostuneita. Toisaalta taas yksilö ei monesti voi vaikuttaa valittuun lähestymistapaan, vaan hänet on määrätty käyttämään sitä. Ja siksi olisi ajateltavissa, että pakosta huolimatta heille kehitty myönteinen suhtautuminen ketteriin menetelmiin.

“Tämä tutkimus muutti halki Nokian mieliä siitä, miten ketterät menetelmät voidaan

nähdä. Ennen tutkimusta puhuimme ketterien menetelmien omaksumisesta – tutkimuksen jälkeen aloimme puhua äänekkäämmin ketterien menetelmien käyttöönotosta – ajatuksena että saatuaan ensikäden tietoa ketteristä menetelmistä ihmiset alkavat ymmärtää niitä paremmin. Tosiasia, että ketterät menetelmät saivat niin myönteistä palautetta, oli yllätys monelle. (s.288)”

Pasi Kiema

Managing software professionals

L. Peters, in Proceedings of Engineering Management Conference IEMC '03, Managing Technologically Driven Organizations: The Human Side of Innovation and Change, pages 61-66, 2003

Johdanto

Ohjelmistoalan alkupäivistä asti ohjelmoijien kustannustehokkuutta ja tuottavuutta on yritetty parantaa erilaisin keinoin. Ohjelmistosuunnittelua ja -tuotantoa varten on kehitetty tekniikoita ja menetelmiä. Ohjelmointityötä on pyritty helpottamaan uusien ohjelmointikielten, työkalujen, ja ohjelmointiympäristöjen avulla. Ohjelmoijien tuottavuuteen liittyvät ongelmat ovat silti yhä ajankohtaisia. Yhtenä syynä ongelmiin pidetään sitä, että ohjelmistotuotannossa vaikuttavat päälliköt ovat yleisesti tehottomia eivätkä osaa huomioida ohjelmoijille tärkeitä asioita.

Tulokset

Ohjelmistoala eroaa monista muista ammateista siinä, että sillä työskentelevät ihmiset ovat taustoiltaan hyvin erilaisia. Koulutustaso voi vaihdella koulunsa keskeyttäneistä tohtoreihin. Koulutus ei välttämättä myöskään ole suoraan ohjelmistotuotantoon liittyvää.

Ohjelmistoammattilaisia tutkittaessa on havaittu kaksi piirrettä, jotka erottavat heidät muiden alojen ammattilaisista. Ensimmäinen on korkea kasvun tarve. Ohjelmoijat yleisesti haluavat haasteita ja tärkeitä työtehtäviä mutta kaihtavat omasta mielestään vähemmän tärkeitä tehtäviä. Toinen havaittu piirre on matala sosiaalisuuden tarve. Ohjelmoijat ovat itsenäisiä ja työskentelevät usein mielellään yksin jonkin ongelman kimpussa. Päälliköiden voi olla vaikeaa luoda ohjelmistoammattilaisista tehokasta ja yhteen hiileen puhaltavaa tiimiä.

Tutkimuksissa on havaittu ADD-tyyppisiä oireita suhteellisen suurella osalla ohjelmoijista. Tyypillisesti ohjelmistoammattilaiset eivät ole tyytyväisiä vallitsevaan tilanteeseen vaan haluavat tehdä muutoksia koodiin, työtapoihin, kehitystyökaluihin ynnä muihin töihinsä vaikuttaviin asioihin. Päälliköiden täytyy osata tasapainoilla riittävän stabiiliuden ja ohjelmoijien muutoshalukkuuden välillä.

Useimmat ohjelmistoalan päälliköt ovat aikaisemmin olleet ohjelmoijia. On helppo ajatella, että päälliköiden ja ohjelmoijien näkökulmat ja arvot olisivat lähellä toisiaan. Näin ei kuitenkaan ole. Ohjelmoijien motivaattoreita ovat aikaansaamisen tunne ja vastuu omasta tekemisestään. Päälliköksi ryhtymiseen motivoi raha ja lisääntyvä valta. Ohjelmoijien arvomaailmassa tärkeimpiä asioita ovat tehdyn työn arvostus, asioista perillä oleminen ja avun saaminen henkilökohtaisiin ongelmiin. Päälliköiden arvojärjestelmää taas hallitsee palkkaan, työn pysyvyyteen ja ylenemismahdollisuuksiin liittyvät asiat. Erilaiset arvomaailmat ja näkökulmat aiheuttavat kommunikaatio-ongelmia ja turhautumista. Päälliköiden on helppo omalla toiminnallaan huonontaa ohjelmoijien suorituskykyä ja motivaatiota vaikka esimiehen tehtävä onkin juuri päinvastainen.

Ohjelmoijien motivointi on tärkeää kahdesta syystä. Ensimmäinen ja ilmeinen syy on kustannukset. Motivoituneet ohjelmoijat tekevät enemmän töitä ja ja työn jälki on parempaa. Toinen, vähemmän ilmeinen syy on myös kustannukset. Tiimissä, jonka työtyytyväisyys on huono, henkilöstön vaihtuvuus on suuri. Jopa 60 prosenttia projektin kustannuksista saattaa johtua henkilöstövaihdoksiin liittyvistä menoista.

Ohjelmistokehityksessä yksi henkilö on useimmiten vastuussa jostain ohjelmiston toiminnallisuudesta tai osa-alueesta. Tällainen omistajuussuhde tarkoittaa sitä, että ohjelmoija ottaa vastuualueisiinsa liittyvät asiat, erityisesti epäonnistumiset, usein henkilökohtaisesti. Ohjelmoijat eivät sitoudu tehtäviin, joissa epäonnistumisen mahdollisuus on suuri. Riskialttiit kehityskäytännöt, kuten hankalat vaatimukset tai liian tiukat aikataulut ovat yleisiä ohjelmoijien motivaatiota huonontavia tekijöitä.

Ohjelmistotuotannossa päälliköiden toimintaan liittyviin ongelmiin on havaittu viisi pääasiallista syytä. Ohjelmistojen abstrakti luonne ja aistein havaittavan työn jäljen puuttuminen aiheuttaa ongelmia ajankäytön ymmärtämisessä. Päälliköiden koulutus on puutteellinen eikä valmista ohjelmistoalan esimiestehtävien haasteisiin. Päälliköt usein luottavat siihen, että tekniikan kehitys ja oikeiden tekniikoiden valinta poistaa ongelmat, vaikka väärinkäytettyinä uudet tekniikat voivat aiheuttaa niitä. Valintakriteerit esimiesasemaan keskittyvät teknisiin taitoihin eivätkä johtamiseen tai strategiseen osaamiseen. Ohjelmistojen julkaiseminen kiireellä on ongelmallista ohjelmoijien kannalta, koska he joutuvat tekemään työnsä hutiloiden ja huonosti, vaikka tietävät osaavansa paremmin.

Ongelmiin voidaan vastata sopivilla käytännöillä. Ohjelmoijien arvomaailmasta johtuen taloudelliset kannustimet eivät ole parhaita motivaattoreita, vaan työn sisältöön ja vaikutusmahdollisuuksiin liittyvät asiat. Kolmea peruseriaatetta soveltamalla ohjelmoijien motivaatioon, työn laatuun ja tuottavuuteen voidaan vaikuttaa positiivisesti. Ensimmäinen periaate on vallan jakaminen. Esimerkiksi aikataulukysymyksissä kehitystiimi voidaan ottaa mukaan pohtimaan sitä, miten työ saadaan tehtyä takarajaan mennessä. Pelkästään päällikkötasolla luodut aikataulut eivät lähes koskaan pidä. Toinen periaate on motivointi arviointien avulla. Esimiestyössä usein jätetään huomiotta palautteen antaminen. Palautetta pitää antaa usein ja työn arvioinnissa täytyy käyttää mielekkäitä kriteerejä. Ohjelmoijalle täytyy myös tehdä selväksi miksi ja miten hänen työnsä on tärkeää yhtiön toiminnan kannalta. Kolmas periaate on työtyytyväisyyteen panostaminen. Ohjelmoijille on tärkeää, että he saavat tehdä laadukasta jälkeä ja tuotteita, joista he voivat olla ylpeitä.

Johtopäätökset tai yhteenveto

Epämotivoivat johtamiskäytännöt aiheuttavat henkilöstön vaihtuvuutta, lisäävät kustannuksia ja huonontavat projektin onnistumismahdollisuuksia. Motivointi ja työtyytyväisyyteen panostaminen lisäävät tuottavuutta ja sitouttavat ohjelmoijia yhtiöön sekä tuotteisiin. Siihen tavoitteeseen sopivien johtamiskäytäntöjen luomiseksi on tärkeää sekä ymmärtää ohjelmoijien motivaatiotekijöitä, että valita ja kouluttaa päälliköt oikealla tavalla.

Jussi-Antti Salomaa