Software project management anti-patterns in innovation projects

Dilem Aydinli

University of Tampere
School of Information Sciences
Computer Science
M.Sc. thesis
Supervisors: Eleni Berki
Timo Poranen
May 2015

University of Tampere
School of Information Sciences
Computer Science / Software Development
Dilem Aydinli: Software Project Management Anti-patterns in Innovation Projects
M.Sc. thesis, 68 pages, 7 index and appendix pages
May 2015

Software project management anti-patterns describe bad practices and their negative consequences in the field of software project management. Anti-patterns also present a refactored solution in order to recover from the negative situation or prevent it beforehand. Anti-patterns are one of the causes of software project failures which have been a significant issue in software engineering and software project management. These failures can be decreased at a certain level with the knowledge of anti-patterns. Therefore, it becomes essential that available anti-patterns in the literature should be studied by project managers, and negative experiences should be faced. Also, knowledge on managing software projects should be shared in an organized way, and new anti-patterns should be investigated.

The aim of this study was to investigate software project management anti-patterns encountered in innovation projects. In order to achieve the purposes of this study, a quantitative research methodology was chosen. A group of university students, who worked as project managers in industrial innovation projects, were requested to fill in a questionnaire. The collected data through the questionnaire were analyzed compared and contrasted to the available anti-patterns in the literature that are mostly found in traditional software projects.

One of the research outcomes of this study is that, some anti-patterns which have been encountered in conventional software projects were also identified in innovation projects. Additionally, a new possible anti-pattern was identified and investigated. The new anti-pattern regards the management of team creativity. In particular, the latter can be considered an anti-pattern because it can be an obstacle in teamwork progress, having negative consequences. A remedy solution was recommended as a re-factoring and recovering management strategy.

This study can be considered as a first step towards a set of practical guidelines for project managers, and innovation project managers in particular; this is a current need that must be addressed. Furthermore, there is a gap of relevant knowledge in the field and there are no similar studies regarding software project management anti-patterns in innovation projects in the literature. Therefore, this thesis could be considered as the first ever research study which investigates and reports on anti-patterns in innovation projects.

Key words and terms: Anti-patterns, software project management, innovation projects.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Assistant Professor Eleni Berki and Lecturer Dr.Timo Poranen for their encouragement, guidance and valuable comments. I would like to thank all the project managers for participating the questionnaire and being a part of this case study.

I would like to thank following people who have directly or indirectly contributed to this thesis: Associate Professor Ioannis Stamelos for sharing his valuable feedback, experience, and time from beginning to end of my thesis, Chara Raptopoulou, who also studied software project management anti-patterns in her master's thesis, for sharing her past experiences and suggestions regarding the case study, Pertti Huuskonen, Kari- Jouko Räihä and Juri Valtanen for their valuable time and comments on the case study while creating the questionnaire.

Finally, I would also like to thank my family for their continued support and motivation throughout this thesis.

Dilem Aydinli
April 2015, Tampere

Contents

List of Figures

List of Tables

1. Introduction

High failure rate in software engineering and software project management have always been a significant issue in the field of software development and project management [Komchaliaw and Wongthongtham, 2010]. Failure rates and its reasons have been investigated by researchers based on various factors: cancellation rate of IT projects, unsuccessfulness rate of delivered projects, project size and so on [El Emam and Koru, 2008]. According to the research of Standish Group in 1994 [Komchaliaw and Wongthongtham, 2010], approximately 31% of corporate software development projects were cancelled before completion and 53% were challenged and cost 180% of their initial estimate [Komchaliaw and Wongthongtham, 2010]. According to another research of Standish Group in 2007 [Komchaliaw and Wongthongtham, 2010], 46% of the projects were not able to fully meet customer requirements and those projects experienced cost and time overruns and 19% of the projects were completely failed [Komchaliaw and Wongthongtham, 2010]. In spite of advancements in development methods and technologies during last decades, software development projects are still prone to fail [Lyytinen and Robey, 1999].

There are several reasons which can lead software projects to partial or complete failure. These reasons can be basically gathered under two viewpoints as external (customer expectations, schedule related issues and so on) and internal. Internal constraints affect basic management practices (planning, team building, decision making and so on) in a negative way [Stamelos, 2010]. However, Brown *et al.* [2000] states, "The primary cause of software development failure is the lack of appropriate project management". Therefore, enforcement of proper and correct software project management functions is an undeniable fact for the success of software development projects.

Software project management is one of the most complex and difficult tasks in software development. Software project management involves management of software development activities with proven engineering management methods, and principles and leadership [Peters, 2008]. In order to be an effective software project manager, one should accomplish both [Peters, 2008]. Additionally, for the case of innovation projects, project managers need to take nature of innovation project into account while managing them since they are more risky cases than conventional projects. While conventional project management is shaped by precision, accuracy, and optimal use of resources, innovation projects include uncertain tasks and objectives [Filippov and Mooi, 2009]. Furthermore, the success and failure in software project management is based on three variables: people, technology and process. As long as all of the variables are managed

successfully, the software development process will thrive. Otherwise, if balance between these variables cannot be addressed, project managers tend to make common classic mistakes to solve certain problems [Brown *et al.*,1998].

Continuous application of common mistakes to commonly occurring problems is known as anti-patterns. Software project management anti-patterns are one of the underlying reasons for failure of a project [Stamelos, 2010]. Management anti-patterns define the key scenarios which give damage to software processes [Brown *et al.*, 1998]. Anti-patterns are a form of patterns. Patterns describe positive solutions to a recurring problem whereas anti-patterns have two solutions as problematic solution which cause negative consequences and refactored solution which make anti-patterns beneficial. According to Laplante *et al.* [2005], "While it is certainly useful to study the successful ways people solve problems, the old adage that we learn from our mistakes suggests that studying failures might be even more fruitful" because anti-patterns provide information regarding the causes of bad practices, negative consequences when bad practices are enforced and a solution to recover from negative situation or prevent it. Anti-pattern knowledge is a good way to learn commonly occurred problems and bad practices in order to solve them, prevent from these bad practices or refactor them by studying others' negative experiences. Therefore, proper documentation of anti-patterns becomes crucial in order to share experiences in this regard. However, available anti-pattern information in the literature is mostly informal and unstructured [Stamelos, 2010]. To get a better understanding of anti-patterns for researchers and project managers, formal and more organized documentation of anti-patterns is a necessity [Stamelos, 2010]. Hence, it is crucial to find out anti-patterns encountered in software development projects. Knowledge on anti-patterns can compromise a set of practical guidelines for software project managers in a way that they can i) learn from others' negative experiences, ii) try to avoid practicing same wrong solutions to the similar problems and iii) apply refactored solutions in order to recover from a negative situation.

1.1. Research questions

Software project management anti-patterns are one of the underlying reasons of software development project failure. Innovation projects are more difficult to manage than conventional projects considering their high risky nature. As stated in the introduction, anti-patterns serve two important purposes; identifying problems, and producing solutions to the identified problems and provide useful knowledge about the commonly enforced bad practices to these specific problems. Therefore, by understanding anti-patterns, software project failure rates can be reduced at a certain level. For this reason, this research will investigate and answer following two questions:

- Which are the software project management anti-patterns encountered in the innovation projects?
- Are there any new software project management anti-pattern(s) that are specific to and commonly found in innovation projects?

1.2. Thesis outline

This thesis is structured as follows. In Section 2, project management and its activities are described in detail. Also, along with project management activities, key management skills that are required to be part of the professional knowledge of an effective project manager are given in this section.

Section 3 goes on describing project management anti-patterns in a rather detailed manner, exposing and outlining necessary background knowledge. This section starts with pattern and anti-pattern relationships and continues with anti-pattern structure and documentation. Project management anti-patterns are described and mostly encountered project management anti-patterns in software development projects are also introduced here.

Section 4 gives information regarding the importance of creativity and project management in innovation projects involving teamwork.

Section 5 consists of the case study, including the motivation and purpose of the case study, environment of the case study, and data collection methodology. Additionally, this section includes collected data, data analysis and results parts.

Section 6 presents the research results, discussion and the conclusions, including limitations of this thesis, and future research recommendations.

The thesis ends with a section on useful references, and also makes available i) the questionnaire disseminated for data gathering, and ii) the ‘raw data’ collected, that is the collection of the immediate answers of the respondents.

2. Projects and their management

Project, project management and project management activities are the core concepts that are necessary to know before leading in project management anti-patterns since the anti-pattern may be the result of not having sufficient knowledge or experience in these subjects and not knowing any better [Brown *et al.*, 1998]. Therefore, having knowledge on projects and their management will ease to understand theory and types of anti-patterns, its negative consequences and solutions to decrease the likelihood or impact of these negative consequences.

In this part, definition of project and project management will be given. Project lifecycle and main project management activities performed during a lifecycle will be explained later on. This part will be concluded with the key management skills which a project manager needs to acquire to present effective project management throughout the course of a project.

2.1. Definition of a project

Organizations are likely to confuse projects with operations since their definitions share many characteristics. Attributes shared by operations and projects are following [Phillips, 2004].

- Performed by employees.
- Have limited resources.
- Go through phases of planning, execution and management.

Although their definitions coincide on a range of attributes, there is some utterly distinctions that separate projects and operations from each other. Despite the fact that many researchers have defined projects in different ways to some extent, according to Project Management Body of Knowledge (abbreviated as PMBOK) which is considered as *bible* of project managers defines a project as "A project is a temporary endeavor undertaken to create a unique product, service or result" [PMI, 2013].

The major difference between operations and projects is that projects are unique and temporary as its meaning implies. By unique, the end result of a project has differences in some distinguishing way from other products of the organization although some repetitive elements exist in some products [PMI, 2013]. For example, even if two software applications which have exactly same requirements need to be built by two different organizations or developers, they can be unique with their design, method, code, programming language, circumstances, contractors, technical resources and so on. By temporary, a project has definite beginning and end dates. A project ends when its

objectives are achieved or it is terminated [PMI, 2013]. Termination becomes necessary when it is not possible to meet objectives or there is no need for the project anymore.

A project has been defined in various ways by different researchers or authors. Juran defines a project as a problem scheduled to a solution [Lewis, 2007]. However, the word meaning of the problem reminds negative situations although projects bring solution to both negative and positive problems. Fairley [2009] defines a project as a coordinated set of activities that occur within a specific time frame to achieve specific objectives. This definition points out an important attribute that a project should follow a life-cycle pattern. According to Pinto and Slevin [1988], a project has to possess beginning and end dates, specific and preordained goals, a series of complex or interrelated activities and a limited budget.

From the given definitions above, it is possible to deduce some of the characteristics of a project. A project can be characterized as beginning and ending dates prescribed, schedule and budget allocated, objectives clearly defined, schedule of activities planned, tasks assigned, work activities coordinated, progress monitored, communication maintained, risk factors are experienced and confronted, and corrective actions applied as necessary [Fairley, 2009].

There are several distinctions between different kind of projects such as manufacturing, hardware, construction, software and other relevant areas. Each kind of projects applies project management procedures to follow development processes and produce an end product as common. There are some essential attributes which separate software projects from other kind of projects and make software projects more difficult such as complexity, conformity, changeability and invisibility of software [Brooks, 1995].

Software projects are complex because of its unique and interactive nature. The reason behind the uniqueness of a software project is that it consists of functions, subroutines and objects which are called when it is needed rather than duplicating them. Also, parts of software interact with each other including serial and concurrent invocations, state transitions, data couplings, and interfaces to databases and external systems. Therefore, software projects can tolerate requirements change to some degree but not completely since a small change in requirements need bigger changes in plan and code [Fairley, 2009].

Software conformity is establishing a software project in compliance with guidelines or specifications. Interfaces or any connection to the environment in a software product is usually designed regarding user specifications and guidelines [Fairley, 2009].

User requirements, dynamic platforms such as hardware, laws, regulations, and objectives are likely to change. Since software products need to keep up with these changes, it can be said that software is changeable [Fairley, 2009].

Software does not have physical properties. Work products such as requirements specifications, design documents, source code, and object codes are not the software itself. These representations of software are used because software has no physical presence. Most of the project failures cause from software invisibility since it is difficult to estimate status of the work product [Fairley, 2009].

2.2. Definition of project and software project management

Project management is "the application of knowledge, skills, tools and techniques to project activities in order to meet project requirements" [PMI, 2013]. A project should go through processes: initiation, planning, execution, monitoring and controlling, and closing in order to present an accomplished project management as it is stated by PMI [2013].

Even though project management is a discipline by itself, it is customized by industries according to their specialization areas. The interest in project management has increased proportionally as business activities have been organized in projects [Söderlund, 2004]. In this thesis, project management tailored by software industry will be discussed.

Software project management is the art and science of planning and coordinating human resources, developing and modifying maintainable software artifacts which meet clients' and users' requirements and expectancy within fixed budget and schedule [Fairley, 2009]. While managing software projects, software project management shares many activities with other sub-disciplines of project management such as identifying requirements, considering stakeholders' needs and expectations in planning and executing the project, keeping in touch with stakeholders frequently, managing stakeholders in meeting project requirements and creating project deliverables, balancing project constraints which include scope, quality, budget, schedule, risks and other relevant sub-disciplines [PMI, 2013].

Software project management is the most challenging one amongst other type of projects because of the nature of software [Fairley, 2009]. Attributes of software, which are complexity, conformity, changeability, and invisibility, cause software to be more difficult as these attributes are explained in Section 2.1. According to Royce [1998], software project management is the most challenging because of its flexibility although flexibility is the best feature of software at the same time. Flexibility feature of software

raises unpredictability in a software project and complicates planning, monitoring and, controlling processes.

2.3. Project life cycle

Project life cycle exists through the course of a project. Life cycle consists of sequential or overlapping phases which have certain start and end dates. Furthermore, these phases can be broke down into milestones, objectives, deliverables, and other related work tasks considering financial condition and scope of the project. In brief, project life cycle defines what technically should be done in each phases [PMI, 2013].

A project life cycle consists of different phases which are commonly shared by most of the projects. These phases are starting the project, organizing and preparing, carrying out the project work and closing [PMI, 2013]. Although there are some common phases through development of different projects, complexity and size of phases are changeable attributes with respect to a project's size and complexity. Level of staff and cost requirements for each phase across a generic project life cycle is shown in Figure 1.

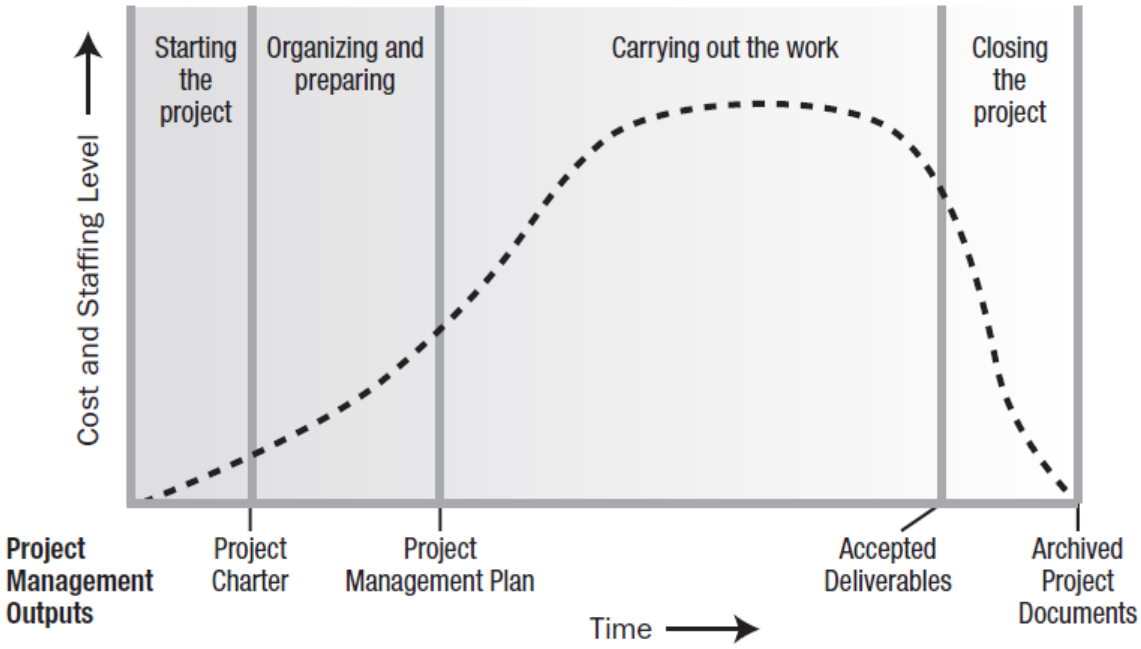

Figure 1. Typical cost and staffing levels across a generic project life cycle structure [PMI, 2013].

Various standardized methodologies are ready for the use of project managers and organizations. It would not be correct to apply same methodology to all kind of projects since each methodology have been developed based on specific kind of projects and various technical, organizational, project and team considerations [CMS, 2005].

For software projects, there are mainly three kinds of life cycle development models as sequential, incremental and evolutionary models [Wallin and Land, 2012]. Sequential development model is also known as traditional development model in which project passes through phases sequentially whereas phases are handled iteratively in evolutionary models. Incremental models remain in between sequential and evolutionary models. These three models are modeled based on the flow of main software development phases: requirements gathering and analysis, design, implementation or coding, component testing, verification and validation testing [Wallin and Land, 2012]. Each phase produces deliverables which will feed into next phase.

In sequential development model, the problem to be solved should be understood completely before passing through design phase. Before implementation, a design which specifies every aspect of the problem should be prepared. All implementation should be done before validation and delivery [Wallin and Land, 2012]. The difference between sequential and evolutionary development model is that all requirements are not finalized in the first phase of evolutionary model. As the old requirements are refined, new requirements are gathered and processed [Wallin and Land, 2012]. In incremental model, like in sequential model, the overall requirements of the project are known from the beginning. However, it can tolerate changes in limited set of requirements [Wallin and Land, 2012].

According to Capers Jones' research which analyses metrics gathered from 15.000 projects, more popular software development methodologies are traditional Waterfall approach, variations of Agile, the Team Software Process (TSP) and Extreme Programming among 55 different development models. In the research, 10 most popular software development models are evaluated. Jones states that many companies adapt most popular development models which today constitute faces of Agile without even evaluating them whether the model fit their projects or not. However, as a result of this research, it is found out that no development model is successful in every size and type of project application. [Jones, 2013]

2.4. Software project management activities

Software project management processes that should be done in particular phases can be grouped under four major activities. Software project managers should accomplish these activities in order to produce a successful software product. Four fundamental activities are given as follows [Fairley, 2009]:

- planning and estimating,
- measuring and controlling,

- communicating, coordinating and leading,
- and managing risk.

2.4.1. Planning and estimating

Since the main objective of every type of project is to complete the project successfully within limited time and resources, the schedule, available resources for the project and effective utilization of these resources should be specified in the project plan. Main planning activities include defining products, defining activities, estimating resources and duration, defining activity network, defining schedule, and total cost. Each of these activities should be applied to a whole project life cycle or a phase. Iteration is important for planning activity because output of later planning activities can be used as a feedback for earlier stages. As a result of feedback of later activities, a revision for earlier activities might be needed. [BSSC, 1995]

In order to begin planning and estimating, project managers need some information about the project. Activities which produce required inputs for planning and estimating activity can be found below [BSSC, 1995].

- Requirements should be gathered from clients and a User Requirements Document should be prepared. According to current project phase, Software Requirements Document or Architectural Design Document should be provided for planning and estimating activity.
- Software product and process standards should be identified.
- Historical data should be acquired for the purpose of estimation of resources and duration.
- Possible risks which can arise through the development of project should be defined, environmental factors such as new technological advancements should be considered.
- Time constraints such as milestones and delivery dates and resource constraints like availability of staff should be documented.

Once required inputs are gathered, project manager can start working on plan and estimation of the project. Software Project Plan is prepared by project managers at the end of planning and estimating. This report contains definition of deliverable products, a process model which defines project development life cycle approach, methods and tools to be used, work breakdown structure which shows hierarchically structured work packages, the project organization which defines tasks assigned to particular team members, an activity network, a schedule of the project including deadline of work packages, a list of required resources for the project and a total cost estimate [BSSC, 1995].

Planning and estimating reveal whether a project results in success or failure. If planning and estimating is faulty, the project will most probably fail [Zwikael and Globerson, 2004]. Therefore, planning is identified as a critical success factor in projects by researchers [Slevin and Pinto, 1988; Zwikael and Globerson, 2004]. Although the importance of planning for a project is already known, there is no attempt to measure quality of planning. However, it is possible to uncover strong and weak points in project planning by measuring quality of planning with proposed diagnostic tools like PMPQ model [Zwikael and Globerson, 2004].

2.4.2. Measuring and controlling

Measuring and controlling activities take output of planning and estimating activities as an input. Progress of the project is measured periodically by evaluating status of particular work activities and comparing them with their planned status. Measurement of project status can be done weekly, bi-weekly or monthly. The relationship of work attributes and status of each attribute are the determinants for project status. Progress reports include information about project status. [Fairley, 2009]

Progress report is kept because it indicates the planned project factors and other ones which need corrective action. These project factors are effort, schedule, cost, progress, product features, quality attributes of the product (e.g. defects, reliability, availability, response time, throughput and relevant attributes), and risk [Fairley, 2009]. For example, personal cost and effort can be controlled with the help of progress report. If effort is as planned and personnel cost is higher, it means that more expensive and highly skilled personnel than planned is assigned to the task. There can be two reasons for this outcome. The work to be done can be more difficult than expected or more skilled personnel's task takes more time than planned. Variation of results from personnel cost and effort measurement can be observed such as planned effort and lower personnel cost, lower effort and cost than planned and so on [Fairley, 2009].

Project control is applied by performing corrective actions when the balance among project attributes are not on the track and progress of the project are not as planned. According to deviation from the plan, corrective actions may vary and include one or more of actions as below [Fairley, 2009].

- Extending the schedule,
- adding more resources,
- using superior resources,
- improving various elements of the development process,
- improving the technology, and
- de-scoping the product.

Measurement and control of software projects is essential in order to provide frequent indicators of progress, provide early warnings of problems, permit analysis of trends for a project, allow estimates for the final cost and completion date of a project and build a data repository of project histories for the organization [Fairley, 2009]. Planning, estimating, measuring and controlling a software project can be considered as practicing institutionalized risk management since these practices increase the probability of a successful outcome and help to find out possible problems that can be encountered through the course of project [Fairley, 2009].

2.4.3. Communicating, coordinating and leading

A project manager is responsible from managing the project and leading the team. However, not all managers can exercise leadership and not all leaders can manage a project. The reason is that there is a clear distinction between managing and leading [Lunenburg, 2011]. Processes of leadership are developing a vision for the organization, supervising people with that vision through communication and motivating people. Leadership produces uncertainty and change in the organization [Lunenburg, 2011]. On the contrary, processes of managing a project involve planning and budgeting, organizing and staffing and controlling and problem solving [Lunenburg, 2011]. Management processes reduce uncertainty and produce predictability in the organization. Summary of the difference between leadership and managing can be observed from Table 1.

	Management	**Leadership**
Creating an agenda	Planning and budgeting	Establishing direction
Developing people to achieve the agenda	Organizing and staffing	Aligning people
Execution	Controlling and problem solving	Motivating and inspiring
Outcomes	Produces predictability and others	Produces change, often useful/dramatic

Table 1. Differences between leadership and management [Mellors, 1996].

One of the primary goals of project managers is to lead their team to foster teamwork atmosphere within a project. In order to foster teamwork, each team member should share same objectives and goals and work cooperatively. If each team member is considered as an individual who has different desires, expectations, goals, and attitudes, team coalescence becomes a serious topic to achieve. Therefore; while forming a team, some aspects should be considered such as personalities of individuals, cultural and social conditions in the organization. Project managers should facilitate some conditions for a team in order to have an effective and efficient team. These conditions can be found below [Fairley, 2009].

- Appropriate number of people,
- correct skill mixture,
- good tools,
- adequate training,
- respect one for another,
- respect for managers and leader,
- willingness to be a team member,
- shared ownership of the work products,
- good communication skills,
- good communication channels,
- good working environment, and
- having some fun together.

Communication is an important aspect not for only teams but also for other stakeholders. A good communication process keeps stakeholders satisfied and teams motivated. Conversely, lack of effective communication increases the risk project encounters and probability that the team will not be able to meet expectations of stakeholders. It is found out that organizations that communicate more effective have more successful projects [PMI, 2013b].

A competent project manager must be good at both managing and leading. Otherwise, he or she has to find out his or her weaknesses and overcome these weaknesses. Additionally, project managers and team members need to know essentials of effective communication since communication is one of the critical success factors of a project.

2.4.4. Managing risks

Project risk is an uncertain event or condition. A risk can affect the project positively or negatively. There can be several factors or only one factor which can cause a risk. For this reason, a risk can impact a project in multiple aspects such as cost, schedule, scope,

quality or performance once it occurs. A cause may be a condition, constraint or given or potential requirement which creates possibility of negative or positive impacts. More explicitly, a cause may be immature project management practices, lack of integrated management systems, concurrent multiple projects or dependency and external participants. [PMI, 2013]

The purpose of risk management is to decrease the impact and likelihood of risks which can affect the project negatively and increase the impact and likelihood of positive events or conditions. Six main activities included in risk management are described by Project Management Institute [2013] as follows.

Plan risk management defines the way to conduct project risk management activities for a project. Project management plan, project charter, stakeholder register, enterprise environmental factors and organization process assets are inputs to the plan risk management process. As a result, a risk management plan is prepared by utilizing tools and techniques such as analytical techniques, expert judgment and meetings. Planning ensures successful risk management within a project when it is done properly. Moreover, planning provides resources and time for successor risk management activities.

Identify risks is the process of investigating the possible risks which may affect the project and document their characteristics. The advantage of this process is to document existing risks and the knowledge and the ability. There are several inputs which must be ready for starting to the process such as risk management plan which is the output of plan risk management, cost management plan, schedule management plan, quality management plan, and human resource management plan and other relevant plans. After applying required tools and techniques which are documentation reviews, information gathering techniques, checklist analysis, and assumption analysis, diagramming techniques, SWOT analysis and expert judgment, risk register is produced. It is an iterative process since new risks can occur throughout the project. Project team should be involved in this process because they can develop and maintain sense of responsibility for the risks.

Perform qualitative risk analysis is the process of calculating probability and impact of risks and prioritizing them considering the calculation result. By this way, project managers are able to cope with high priority risks and reduce their impact and probability. In order to perform qualitative risk analysis, project managers need risk management plan, scope baseline, risk register, enterprise environmental factors and organizational process assets. Then, project managers asses risk probability and impact, risk data quality and risk urgency, create risk probability and impact matrix, categorize

risks with the help of an expert judgment. As a result of these sub-processes, project documents update is produced. Perform qualitative risk analysis is regularly performed throughout the project life cycle. It is a rapid and cost effective way of establishing risks for risk response and lays the base of performs quantitative analysis.

Perform quantitative risk analysis is the process of quantitatively analyzing the effect of identified risks. The benefit of performing quantitative risk analysis process is to reduce project uncertainty by providing quantitative data for project risks. Inputs to this process are risk, cost and schedule management plan, risk register, enterprise environmental factors and organization process assets. Project managers should apply data gathering and representation techniques, quantitative risk analysis and modeling techniques and expert judgment to be able to update project risk documents. Perform quantitative risk analysis should be repeated until risks are satisfactorily decreased.

Plan risk responses is the process of developing options which reduces the impact and probability of negative risks and enhance positive risks on entire project objectives. Its advantage is to address risks by their priority and insert resources and activities into the budget, schedule and project management plan as needed whenever inputs, risk management plan and risk register are required. Project managers need to develop strategies for negative risks, positive risks and contingent response. Expert judgment is an essential part of plan risk responses process. Project plan and project documents are updated as a result. Risk responses should be appropriate for the significance of the risk, cost-effective, realistic within the project context, agreement of other parties and owned by a responsible person.

Control risks is the process of implementing risk response plans, tracking identified risks, monitoring residual risks, identifying new risks, and evaluating risk process effectiveness throughout the project. Advantage of the process is to optimize risk responses by improving efficiency of risk approach. Project management plan, risk register, work performance data and work performance reports are required inputs for this process. Tools and techniques used in this process are risk reassessment, risk audits, variance and trend analysis, technical performance measurement, reserve analysis and meetings. Once the process is completed, work performance information change requests, project management and document updates and organizational process assets updates are produced. [PMI, 2013]

An organization should conduct risk management activities consistently and proactively in order to be successful. Identifying and maintaining effective risk management throughout the project lifecycle can be beneficial for all levels of an organization for better decision making. Project risks could occur even in the initiation

phase. Without effective risk management, problems may increase because of unmanaged threads. [PMI, 2013]

2.5. Project manager and key management skills

A project manager is the person who is responsible for achieving the project objectives in the leadership of a team. Key project management responsibilities include creating achievable and clear project objectives, requirement and constraint management which concerns time, scope, quality and cost attributes [Wikipedia, 2014]. A software project manager should have deep knowledge in software development which includes programming, documenting, testing, application development and other relevant areas. This knowledge is usually acquired by experience and education. Moreover, it is expected from a software project manager to be familiar with software development lifecycle [Wikipedia, 2014]. In addition to any area specific knowledge and general management proficiencies required by the project, the project manager should obtain skills: knowledge, performance, personal, communication, organizational, team building, leadership, coping and stress management, decision making process, political and cultural awareness, negotiation, conflict management and coaching for effective project management [PMI, 2013].

Knowledge refers to project manager's knowledge about project management and performance refers to project manager's capabilities while practicing his or her project management knowledge.

Personal refers to project manager's behavior during project development. Project managers work in a goal driven and time sensitive environment through the project team and stakeholders. A balance between interpersonal, ethical and conceptual skills is required in order to be an effective project manager. Some of the key interpersonal skills are described below.

Communication is an essential skill which brings project managers to success. Effective communication within team and other stakeholders is crucial [PMI, 2013]. Being persuasive and having listening skills are characteristics of a good communicator in project management context [Meredith and Mantel, 2009].

Organizational skills include planning and goal-setting abilities along with the ability to be analytical [Meredith and Mantel, 2009].

Team building is an important part of teamwork. Essential team building skills include developing emphatic relationship with team members, caring about needs of the team, motivating people and strengthening the team spirit [Meredith and Mantel, 2009].

Leadership skills involve setting a good example, seeing the big picture, being enthusiastic and energetic, having positive outlook, taking initiative and trusting people [Meredith and Mantel, 2009].

Coping and stress management skills are required for successful project management since change is a constant in projects. While coping with changes, flexibility, creativity, patience and persistence are the expected skills from a project manager [Meredith and Mantel, 2009].

Decision making process may involve project managers individually or project team. Project managers use four basic decision making styles as command, consultation, consensus, and random. There are four major factors that affect decision making style such as time constraints, trust, quality and acceptance [PMI, 2013].

Political and cultural awareness becomes essential for project managers when the project team consists of people from different cultures and background. By understanding cultural differences, project managers should use politics and power. In such a case, these skills help project manager to be successful and build mutual trust environment [PMI, 2013].

Negotiation is an integral part of project management. When negotiation is truly utilized, it increases the probability of project success. Negotiation is confronting parties which do not share same interests in order to reach an agreement [PMI, 2013].

Conflict management can turn conflicts into an advantage since possible risks can be prevented in addition to settling the conflict with a good practice of conflict management. Thus, project managers should develop themselves to obtain related skills for coping with conflicts [PMI, 2013].

Coaching is to develop team’s competency and performance to higher levels. There are different forms of coaching such as formal and informal training. By helping teams to develop their abilities and skills, team members become more motivated and confident. Therefore, they can present more productive work as a team [PMI, 2013].

3. Project management anti-patterns

This section explains bad practices, in other words anti-patterns, in project management. Firstly, pattern and anti-pattern relationship is described since an anti-pattern is a form of a pattern. Then, anti-pattern definition, its root causes and primal forces will be explained. Anti-pattern documentation is an important process for sharing bad practice experiences with others. However, while documenting anti-patterns, a common template should be used in order to keep the report straight, prevent duplicates and share sufficient information. Therefore, anti-pattern templates and documentation will be discussed next. This part will be concluded with the introduction of common project management anti-patterns encountered by project managers.

3.1. Patterns and anti-patterns

A pattern can be defined as

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" [Alexander *et al.*, 1977].

This general definition of a pattern is also valid for design patterns. A solution is referred as a pattern after constant observation of patterns in practice. In general, a pattern has four fundamental elements [Gamma *et al.*, 1994]:

- *Pattern name* puts a design problem, its solutions and consequences in a nutshell and describes them in one or two words.
- *Problem* describes in which situations a pattern must be applied. The context and problem which can be solved with the implementation of a particular design pattern are explained.
- *Solution* explains relationships, responsibilities and collaborations of general elements of a pattern which solves a design problem.
- *Consequences* are the results and consequences of a pattern once it is applied.

Besides four fundamental elements of patterns, more detailed templates for patterns include more information such as implementation details and the scale of the pattern while documenting those [Gamma *et al.*, 1994]. Patterns became very useful in design level and thus software engineering has started to utilize design patterns. Design patterns have a wide area of utilization in software development projects such as computing, concurrency, distribution, performance, and scalability [Settas, 2011].

The role of patterns is to ensure positive solutions to a commonly occurring problem, whereas anti-patterns have two solutions as producing problematic solutions which cause negative consequences and difficulties and refactored solutions which make anti-patterns beneficial. Refactored solutions provide means for the project and direct the project towards success [Kärpijoki, 2001; Brown *et al.*, 2000]. Anti-patterns are closely related to design patterns since design patterns can evolve into anti-patterns as it can be observed from Figure 2. When Figure 2 is examined, it can be seen that design patterns have two essences as a problem and a solution. A problem can be varied according to its context and design forces. Besides, a pattern documents a repeatable solution to the associated problem. A solution generates benefits, consequences and related solutions. On the other hand, an anti-pattern presents a solution commonly applied to a particular problem which generates negative consequences. The anti-pattern provides information about the reason of the wrong practice applied to a specific problem and shows how to prevent and correct the solution. Because of the pace of technological advancements, technology is changing rapidly. Therefore, patterns are transforming into anti-patterns over time. That's why the number of existing anti-patterns is more than patterns today [Settas, 2011; Brown *et al.*, 1998].

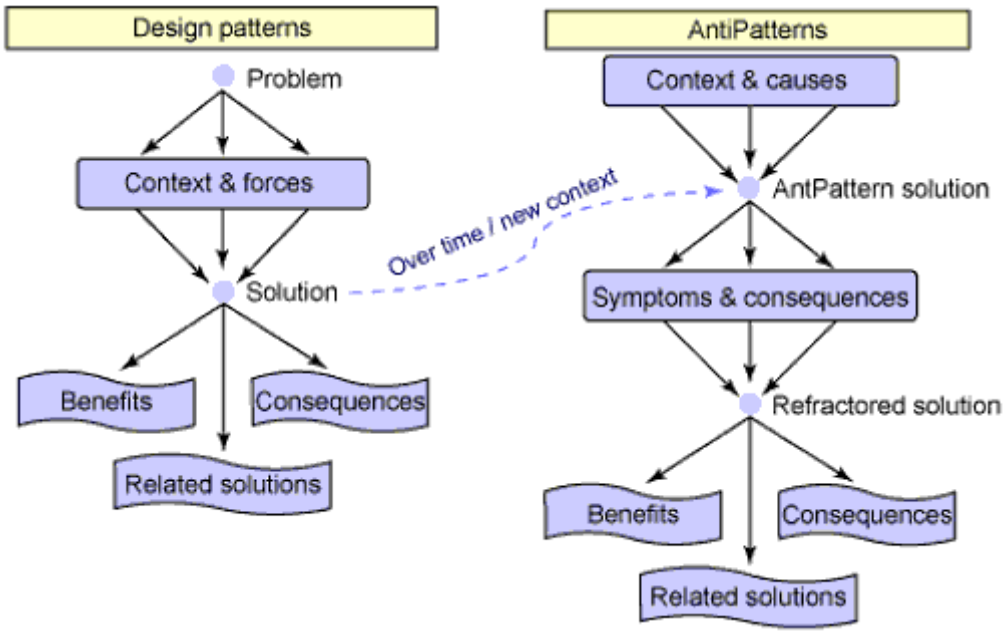

Figure 2. Design patterns and anti-patterns [Settas, 2011; Brown *et al.*, 1998].

3.2. Anti-patterns

There has been an increase in publications about design patterns since 1994. Various design patterns were provided to software architects to evaluate and apply them in object oriented software projects. This popularity in enormous utilization of design

patterns brought good and bad sides. Reusable design patterns were produced and many seminars and articles about design patterns were prepared to increase software project workers' awareness. However, bad side of design patterns created anti-patterns because of the misusage of design patterns. [Brown *et al.*, 1998]

Discussion of anti-patterns started with the introduction of design patterns correspondingly. Before the use of actual word anti-pattern, dysfunctional behaviors and refactored solutions have been documented by Brooks [1979], Webster [1995], Coplien [1994], and Andrew Coenig [Brown *et al.*, 1998]. The word anti-pattern was formally used in a conference entitled "Anti-patterns: Vaccinations against Object Misuse" by Michael Akyord [Akyord, 1996; Brown *et al.*, 1998]. Since anti-patterns have so many contributors, it would be wrong to grant its patent to a single source. Brown [1998; 2000] who made a great contribution to the anti-patterns defines anti-patterns as below.

"An anti-pattern is a literary form that describes a commonly occurring solution to a problem that generates decidedly negative consequences" [Brown *et al.*, 1998]. Anti-patterns may be caused by managers or developers. They may not have sufficient knowledge or experience to solve a particular problem, not know any better or apply a design pattern in a wrong context [Brown *et al.*, 1998]. To call an anti-pattern is an anti-pattern, it must be witnessed at least three times [Laplante and Neill, 2006].

It is possible to categorize anti-patterns under four viewpoints as:

- *Development anti-patterns* include technical problems and solutions encountered by programmers while solving problems.
- *Architectural anti-patterns* identify and resolve common problems regarding structure of systems.
- *Managerial anti-patterns* are related to the problems about software processes and development organizations. [Brown *et al.*, 1998]
- *Environmental anti-patterns* describe problems caused by a prevailing culture or social model. Misguided corporate strategy or uncontrolled socio-political forces are the main reasons of environmental anti-patterns [Laplante and Neill, 2006].

A reference model which introduces the key concepts of anti-patterns is based upon three aspects as root causes, primal forces and software design-level model (SDLM). Root causes are the common and fundamental underlying reasons for problems. Primal forces are the motivators involved in decision making. The relative significance of these forces is determined by architectural scales defined by SDLM. [Brown *et al.*, 1998]

3.2.1. Root causes

Root causes are the mistakes which bring software projects to failure. Root causes' names are inspired from seven deadly sins as follows.

- Haste – quickly made decisions in software design and implementation lead to compromises in software quality. Many project deadlines and milestones are missed because of the unrealistic targets in a project schedule. Therefore, the schedule has to be slipped to catch close of the project. For this reason, testing phase is skipped in most cases. Beside low quality software, security gaps and bugs in the code can be found as a result of non-tested software. Especially large projects which apply Waterfall model call hasty decisions due to the fact that it is difficult to estimate the schedules from the beginning [Brown *et al.*, 1998; Kärpijoki, 2001].
- Apathy – although there is a known solution for a particular problem, apathy is the unwillingness to attempt a solution. It is caused by unhealthy team relationships and poor human relationships between team and managers. Apathy leads to ad-hoc designs and confusing architectural choices using object oriented and techniques of distributed systems. Reusable and replaceable designs are ignored in design and implementation [Brown *et al.*, 1998; Kärpijoki, 2001].
- Narrow-Mindedness – solutions which are widely known are not practiced despite their effectiveness and benefits are perceived [Brown *et al.*, 1998]. Narrow-mindedness leads to poor code with undocumented hard-coded definitions, ad-hoc solutions and complicated design in software development level [Kärpijoki, 2001].
- Sloth – poor technical and managerial decisions are made because of laziness of a manager or a developer. Their approach to a problem is the easiest way only considering near future. These decisions may obstruct the development process in the long run [Brown *et al.*, 1998; Kärpijoki, 2001].
- Avarice – excess complexity in design and implementation is observed in this root cause. Avarice leads to challenges and problems in a software project since excessive systems are very expensive to develop, integrate, test, document, maintain and extend [Brown *et al.*, 1998].
- Ignorance – ignorance is intellectual sloth. Sloth can be encountered in both managerial and developmental level. If a developer is ignorant and tries to find a quickest possible solution to a problem by escaping from the reality, hard-coded elements of code, ad-hoc solutions and poor interface level design can be caused. If a manager is ignorant, potentially severe problems are ignored and

decisions to be made are postponed to another time. As a result of ignorance in managerial level, catastrophic effects can be observed. [Brown *et al.*, 1998]

- Pride – developers implement features which will never be required and invent new designs even if there is already existing ones. These practices lead to unnecessary risk, usage of resources and cost. Pride in managerial level makes a project manager to specify unrealistic requirements which are impossible to implement [Brown *et al.*, 1998].

3.2.2. Primal forces

Forces are concerns and issues included in decision-making processes. Unresolved and successfully addressed forces can affect the solution in different ways. There are two types of forces as vertical and horizontal. Vertical forces are domain specific and unique to one software situation whereas design choices can be applicable across several software modules in horizontal forces. Primal forces are certain architecture and development related horizontal forces because they affect architecture and development processes and appear in most decision making processes [Kärpijoki, 2001]. Different primal forces can have relative important effect at particular scales such as global industry, enterprise, application and system levels. Primal forces from the lowest layer to highest layer can be found below.

Management of functionality is that developers and system architects have to ensure that all requirements elicited from the clients are met within the associated software product. It affects the application level incontrovertibly [Brown *et al.*, 1998]. However, there can be changes in requirements or all requirements cannot be gathered from the customer initially. In such cases, management of functionality cannot be feasible [Kärpijoki, 2001].

Management of performance's effects are critical in both system and application levels [Brown *et al.*, 1998]. Management of performance aims to meet performance requirements despite functionality and performance requirements are contradictory aspects. Performance estimations can be difficult to perform during development of new products since the main concern is implementing all of the features to the system. When the implementation phase is over, during test phase it can be observed that performance of the product is below expectations [Kärpijoki, 2001].

Management of complexity is important for global industry, enterprise, and system and application levels. The significance of software abstraction cannot be denied in terms of reusable and independent components. Abstraction provides simpler interfaces, uniform architectures and improved object models. However, lack of abstraction in a software product leads to system complexity. [Brown *et al.*, 1998]

Management of change controls adaptability of the system [Kärpijoki, 2001]. It is desired to have complete requirements initially although it is very hard to fulfill this desire. Therefore, changes should be tolerated by software management, architecture and development processes. Otherwise, endless bureaucracy, overly complex and messy architecture and implementation can be emerged. Since it is important to make adaptable systems to any change, management of change influences enterprise and system level most. [Brown *et al.*, 1998]

Management of IT resources concerns the ability to manage technical and use of human resources. The management of IT resources includes hardware/software acquisition, inventory, training, maintenance, upgrade and support. Management of IT resources' effect can be seen in enterprise and system level. [Brown *et al.*, 1998]

Management of technology transfer involves formal and informal relationships which the enterprise and industry can share their knowledge about software and technology. At the present time, it is very easy to transfer knowledge globally due to the Internet. Management of technology transfer is important at the global industry level because of reuse of knowledge and experience. [Brown *et al.*, 1998]

3.2.3. Software design-level model

A system cannot tolerate requirements and technological changes if it is developed on a piecemeal basis without an overall architecture. The most important benefit of architecture is partitioning a problem into solvable parts instead of considering and tracking a system as a whole. Software design level model consists of seven levels. These levels are presented from the finest grain level to the largest global level as follows [Brown *et al.*, 1998].

- Object level – in this level main concern of a software developer is management of object classes and instances. Its goal is to build basic functions included in application requirements. Secondary goals include reuse of existing software and documentation.
- Micro-architecture level – combinations of multiple objects and object classes are provided by using patterns. The level mainly concerns small designs which can solve specific problems.
- Framework level – a framework utilizes multiple design patterns at the macro component level which consists of micro-architectures.
- Application level – application level consists of one or more framework level and design patterns. The main goal is to implement a set of functionalities which compromise with the software requirements.

- System level – it consists of several integrated applications. System level provides communication and interoperation between applications.
- Enterprise level – enterprise level compromises several systems which make this level largest architectural scale within an organization. The enterprise identifies policies and procedures in order to keep technical and human resources under control.
- Global/Industry level – global level compromises several enterprises. In this level, de facto standards are defined which are used by organizations. [Brown *et al.*, 1998]

3.3. Anti-pattern structure and documentation

Pattern templates are very crucial in order to provide consistent way to describe patterns and anti-patterns [Jimenez, 2006]. A template can answer essential questions regarding patterns in a pattern language, pattern catalog, or a pattern system [Brown *et al.*, 1998]. While documenting design patterns, graphical notations which represent the relationships between objects and classes are not enough. In order to provide a reusable design, decisions, alternatives and tradeoffs that led to it need to be considered. Although there are different ways to formally or informally describe a pattern, a strictly defined formal template which describes a design pattern includes pattern name and classification, intent which explains the purpose of using a particular pattern, also known as, motivation which includes a scenario about a design problem and how a pattern can solve it, applicability of the pattern, structure which is the graphical representation of the classes in the pattern, participants which are basically classes and objects and their responsibilities in the pattern, collaborations of participants to fulfill their responsibilities, consequences of using the pattern, implementation which includes pitfalls, hints or techniques while applying the pattern, sample code, known users which consist of at least two examples from different domains and related patterns. [Gamma *et al.*, 1994]

Templates are the one of the best ways to define differences between patterns and anti-patterns. The reason is that patterns have one solution whereas anti-patterns have two solutions as a solution which generates negative consequences and a refactored solution which includes improved benefits and decreased consequences [Brown *et al.*, 1998]. Furthermore, solution of a pattern must conform to the rule of three occurrences. According to this rule, a real world solution should have been applied at least three times in order to call a pattern as a pattern [Cunningham and Cunningham, 2011]. However, the situation is different in the case of anti-patterns. Although the first solution of an anti-pattern must conform to the rule of three occurrences, it is not likely

to experience that known occurrences of the anti-pattern were solved in exactly the same way [Brown *et al.*, 1998].

There are several ways to describe an anti-pattern using various templates. Pseudo-Anti-pattern template includes information about the name and definition of the problem. This is not a useful template since it does not really include sufficient information about the anti-pattern. Mini-Anti-pattern template is an informal way to describe an anti-pattern. The template consists of the name, description of the problem and refactored solution as the nature of anti-patterns require two solutions unlike patterns. These two templates are informal and do not really cover all structures of an anti-pattern. Besides of these, full anti-pattern template comprehensively and formally describes an anti-pattern. Aspects considered in full anti-pattern can be analyzed below [Brown *et al.*, 1998].

- *Name* is a unique identifier for an anti-pattern which is descriptive and memorable.
- *Also known as* includes other popular names which describe an anti-pattern.
- *Most frequent scale* is the original scale of an anti-pattern. A SDLM layer should be identified where an anti-pattern is applicable most (see subsection 3.2.3).
- *Refactored solution name* identifies the refactored solution pattern.
- *Refactored solution type* is the methods to obtain desired solution. The type of refactored solution refers to the actions regarding technology, development processes, human resources and software.
- *Root causes* are the causes of an anti-pattern (see Subsection 3.2.1).
- *Unbalanced forces* define misused primal forces of an anti-pattern (see Subsection 3.2.2).
- *Anecdotal evidence* is the phrases associated with the anti-pattern which help to memorize easily. This is not a compulsory section.
- *Background* contains general background information which can be useful and some examples of where problems occur. This is also an optional section.
- *General formex* plains general characteristics of an anti-pattern with a diagram.
- *Symptoms and consequences* are the outcome result from an anti-pattern.
- *Typical Causes* identifies unique causes of an anti-pattern.
- *Known expectations* cover behaviors and processes which are acceptable.
- *Refactored Solution* is the solution which makes an anti-pattern beneficial and resolves the forces in the anti-pattern.
- *Variations* include known major variations and alternative solutions of the anti-pattern.

- *Example* explains how to apply refactored solution to a particular problem identified by an anti-pattern.
- *Related Solutions* present related anti-patterns and their commonalities and differences.
- *Applicability to other viewpoints and scales* defines managerial, architectural or developer viewpoints which are impacted by anti-patterns.

There are optional and required parts to be filled in this template. For example, general forms of an anti-pattern and refactored solution part have to be filled. Although full anti-pattern template is the detailed one which includes extensive anti-pattern structures, it has not been used in the literature because one should have background knowledge on some specific parts [Settas, 2011]. Alternatively, there are different kinds of templates for documenting anti-patterns and they include common structures of anti-patterns. As it can be observed from Table 2, Laplante's [2006] customized template consists of name of the anti-pattern, central concept, dysfunction, explanation, band-aid, self-repair, refactoring and identification.

Name	**A name that conveys the meaning of an anti-pattern**
Central Concept	A short synopsis of the anti-pattern.
Dysfunction	The problem with the current practice with a list of symptoms.
Explanation	Expanded explanation of the anti-pattern including its causes, consequences and historical or cultural analogies.
Band-Aid	A short term fix or coping strategy for those who have neither the time nor influence to refactor it properly.
Self-repair	The first step for someone perpetuating the anti-pattern.
Refactoring	Description of the changes that should be made to remedy the situation and their rational.
Identification	An assessment instrument consisting of a list of questions for diagnosis of the anti-pattern.

Table 2. Laplante's anti-pattern template [Laplante & Neill, 2006].

3.4. Project management anti-patterns

The recent and frequent problem with modern engineering profession is that it involves human communication and resolving people issues. Management anti-patterns define the key scenarios which give damage to the software processes [Brown *et al.*, 1998].

Anti-patterns are revolutionary approach to software development [Brown *et al.*, 2000]. The purpose of anti-patterns is to help project managers to create awareness which brings forth enhancement in their success [Brown *et al.*, 1998]. Anti-patterns assist project managers in producing refactored solution when things get worse due to the fact that anti-patterns are leveraged from the principles of patterns to provide a good solution, working practice and learning from others' experiences [Brown *et al.*, 2000].

Project management anti-patterns are caused by an individual, a management team or executives [Laplante and Neill, 2006]. The major variables in the failure and success of a project are people, process and technology. As long as all of the variables are managed successfully, the software development process will thrive. Otherwise; if balance between these variables cannot be addressed, common classic mistakes can be made by project managers [Brown *et al.*, 2000]. That's why; anti-patterns are based on failures in these critical areas. Some of the prevalently encountered project management anti-patterns, which are explained in detail in Brown's [1998] book, are given with the anti-pattern problem and the refactored solution below.

3.4.1. Analysis Paralysis

Problem

Analysis Paralysis anti-pattern occurs when the project manager strive for completeness and perfection in the analysis phase. Project managers feel more confident in their ability to analysis and decompose than design and implementation. Furthermore, they want to complete all analysis before design phase begins. Moreover, goals in the analysis phase are not well defined. Due to these reasons, the project cannot go further than the analysis phase. Detailed and complex analysis makes the project difficult to develop, document and test. There can be increase in cost of analysis than expected.

Refactored Solution

Instead of using sequential development models, an incremental model can be applied to the project as a development model. By that way, project managers do not have to complete all of the tasks in their associated phases. Details of the problem and its solution can be understood throughout the course of the project by visiting each phase multiple times.

3.4.2. Blowhard Jamboree

Problem

So called industry pundits disseminate marketing information which concerns customers and influence technology decisions. Popular media and private publications publish controversial reports which involve information about particular technologies. Developers spend too much time to answer concerns of project managers arisen from such reports in addition to their real tasks in the project. Therefore, these so called industry experts are misinformed and represent prejudiced information.

Refactored Solution

In such a case, significance of an in-house expert on each key technology is understood better. The expert will be able to distinguish the facts between misinformation, opinions shared in social media and confidential reports. As an alternative solution, staff employees for developing their expertise in particular technologies by reading, attending seminars and other relevant activities.

3.4.3. Corncob

Problem

Corncobs are the difficult people who prevent others from their work and create a negative working environment. The reason behind their behavior would be their personality. However, this attitude often arises from lack of motivation, monetary incentives or recognition.

Refactored Solution

There are several ways to deal with corncobs such as strategic, operational and tactical ways. Tactical solutions are employed on the fly. Operational solutions are applied in private meetings with the troubled person. Strategic solutions are extended over a period of time. In this regard, concord support group can be formed if an organization suffers from many corncobs.

3.4.4. Death by Planning

Problem

Death by planning occurs because of excessive and detailed pre-planning of software projects. Many projects fail from over planning. There are two types of over planning as Glass Case Plan and Detailed Plan. Glass Case Plan is caused by outdated plans which are considered as accurate plans over the course of project development. Detailed Plan

is caused by plans which include very detailed and unnecessary information on the purpose of effective delivery. As a consequence of this anti-pattern, late delivery of the project can be in question, crisis in project management can occur, and cancellation of the project can be thought.

Refactored Solution

As a first step of the solution, deliverables should be shown in the project plan. Deliverables plan need to be updated weekly to ensure appropriate planning and controls which reduce project risks. Deliverables can be presented in a Gantt chart with their deadline and interdependencies. Therefore, state of deliverables (as on scheduled, delivered, early or late) can be tracked easily. Additionally, it is advised to have some flexibility in schedule for unexpected changes.

3.4.5. E-mail is Dangerous

Problem

Although e-mail system provides quick and easy way of communication, it is mostly inappropriate for many topics and sensitive communications. E-mail debates can easily end up with disagreement. Morale and productivity of the project can decrease because of the conflict caused by e-mail discussions.

Refactored Solution

E-mail should not be used for sensitive, controversial, or confrontational messages. Face to face discussions and telephone conversations is less likely to damage the working atmosphere and productivity of the project.

3.4.6. Fear of Success

Problem

When the project is close to the successful completion, some people can begin to worry obsessively about the things that can go wrong. In order to ease their concerns, they can make irrational decisions and act inappropriately. This situation can lead to negative advertisement of the product outside the project team and consequently project outcome can be influenced adversely.

Refactored Solution

In such a situation when the project is near to successful completion, *declaring success* would be a good solution. By that way, project workers accept the significance of the work they have done and appreciate their accomplishments. Termination of the project

can also be mitigated by declaring success. Moreover, project workers' commitment to the organization can be deepened and it can lead up to the future project activities.

3.4.7. Irrational Management

Problem

Project managers are not able to make decisions because of their personality or too much obsession with the details. The anti-pattern is significantly compounded by a manager's inability to manage development staff, other managers and development processes. Those project managers also do not have clear vision and strategy. As a result, they cannot make decisions, can fear success and can ignore true state of the project deadlines and deliverables.

Refactored Solution

In order to solve such a possible problem, a guideline can be followed as below.

The project manager

- Needs to admit that he or she has a problem and gets help.
- Should understand the personality of each team members in addition to their technical capabilities.
- Should set clear and short term goals which should be easy to understand and stimulate team members' motivation.
- Should share the focus of the project with team members.
- Should control and monitor development of the project.
- Should facilitate communication and identify key concepts and achievements of a debate.
- Should manage communication techniques. She or he should not allow team members to have electronic debates and should prevent miscommunication by tracking e-mails.
- Should allow problems sometimes to be resolved by others by managing by exception without interfering.
- Should apply effective decision making techniques.

3.4.8. Project Mismanagement

Problem

Project mismanagement is caused by lack of control and monitor of the project processes. Key activities in a project are minimized such as architectural planning, inspection and test activities. For this reason, inspection, testing, integration and

interoperability cannot be done properly because of missing technical criteria in inadequate architectural plan. Since code reviews and inspections are rarely applied, it is not possible to find design defects.

Refactored Solution

Proper risk management plan should be applied in order to resolve consequences of project mismanagement anti-pattern. Risks are categorized as managerial, common project failure points, quality and common understanding. A project manager should understand these categories to apply better risk management plan to the project.

3.4.9. Smoke and Mirrors

Problem

A product demonstration is one of the best sales tools if the product is high quality. Project manager can sometimes make commitments for the product even if these promises are beyond the organization's capabilities and it is impossible to be implemented by developers in a specified time. When end users cannot see what they are promised and expect in the demonstration, they can be frustrated by the unqualified product. Therefore, promotion of the product can be damaged and development organization can lose credibility.

Refactored Solution

Project managers should manage expectation of end-users considering ethical reasons. It is better to let people expect less than the product to be delivered. End-users will be very satisfied with the result when the product is beyond their expectations.

3.4.10. The Feud

Problem

Managers who have serious conflicts with other managers influence the working environment negatively and employees are the ones who primarily suffer from managers' disagreements. Corporate productivity and image can be damaged as a result of animosity between managers.

Refactored Solution

Friendly office gatherings should be arranged and organizational problems should be solved during these gatherings. Also, this kind of activities are beneficial for team building, socializing and encouragement the development of cross organizational facilities.

3.4.11. Throw It over the Wall

Problem

Although documentation is rarely self-explanatory, personal insight and vision of authors which drives them to make decisions are very important factors to understand the documentation. Understanding insight of authors helps to understand their work better.

Refactored Solution

The material included in the documentation should pass through several important techniques in order to understand and implement the technical documentation as intended. One of the ways is to transfer information by conducting tutorials. It is found out that one day session is enough for development specifications up to 100 pages. Providing two different sessions as managerial and development would give better result.

3.4.12. Viewgraph Engineering

Problem

Developers substantially produce viewgraphs and documents instead of developing software. Because of organization's technical incapability, they are forced to use office software tools to produce technical diagrams and papers which prevent them from using their abilities and cause frustration in development team.

Refactored Solution

Prototypes and mockups can be utilized as a part of system development process. Prototypes provide technical information which cannot be acquired from paper analysis and reduce technology and user acceptance related risks.

4. Software innovation projects

Innovation projects involve not only product and service innovation, but also various aspects of innovation and innovativeness. Therefore, innovation projects should be comprised of at least one of the criteria given below [Filippov and Mooi, 2009].

- Product innovation – purpose of the innovation should be development of a new product or service.
- Process innovation – it is the application of innovative approaches and methods.
- Organizational innovation – improvement of innovative and learning capabilities of project executer accelerates thanks to an innovation project.
- User innovation – users have close interaction with project owner.

The intensity of innovation can vary from one project to another. There are basically three intensity degrees of innovation: imitation, incremental innovation and radical innovation [Filippov and Mooi, 2009]. Imitation is the application of an existing product in the market which is new to the customer. Incremental innovation aims to improve an already existing product at a considerable degree. The purpose in radical innovation is to build a unique product. As the intensity of innovation increases, the degree of technical, market and organizational uncertainty increases proportionally [Filippov and Mooi, 2009].

4.1. Innovation and creativity

Innovation and creativity have several definitions. Creativity simply means the act of producing new ideas, approaches or actions [Addis, 2009] whilst innovation can be defined as the sifting, refining and most basically implementation of those ideas [Gurteen, 1998]. From the definitions of both terms, it is apparent that they are interrelated notions.

Innovation takes new or existing ideas and puts them into action with the application of existing knowledge and development of new ideas. Therefore, new ideas have an important role in innovation [Gurteen, 1998]. Correspondingly, it can be said that innovation springs from the creative application of knowledge. Thus, innovation has two important components as a stock of knowledge and creativity which is the keystone of innovation. Often an invention is the first several stages before an innovation is fully realized [Yusuf, 2009].

Innovation originates from creativity as mentioned above. There are some distinctions between people who are gifted for creative innovations between others as it can be seen below [Addis, 2009]. Expertise, creative thinking skills and intrinsic

motivation are some distinctions. Expertise is specialization in a particular discipline and creative thinking skills gives flexibility and imagination which are related to problem solving skills.

Creative interest and the degree of creativity are mostly influenced by culture and traditions. The ratio of creative people in a society can be increased once the society fulfills the circumstances as described below [Yusuf, 2009].

The society puts a premium on learning and promotes talent through cultural reinforcements and nurturing family environment from early childhood to the formative years. As proved by Trends in International Math and Science Study (TIMSS), when high quality education is supported by the cultural environment of home, progress can be made in cognitive and analytical capabilities and motivation which are important for scholastic performance. Organizations seek employees with better problem solving skills, capacity to convey complex information and ability to work in a group environment effectively since technological advances and computerization enhances dramatically. Therefore, better education becomes a priority in this regard. Better education can be provided by utilizing technological devices in a classroom environment and well-trained teachers.

A creative society pays attention to building human capital by ensuring better health beginning from childhood when good nutrition and care have effective and persistent implications for learning capacity in later life. Physical health and mental well-being are two essential components for creativity. Happy and healthy people are presumably more productive, pursue knowledge more passionately and put the knowledge into practice in a more creative way. In other words, creativity will be influenced by the physical, emotional, and intellectual quality of human capital. In consequence, investment in human capital should be made in order to achieve creative society.

Students and researchers must adequately master a narrow subfield to be able to enhance the frontier of knowledge with the knowledge growing at an exponential rate. There are two implications of this condition. The first one is that individuals were making their first explorations in later ages in the past. In such a case, the productivity in innovation can decrease, especially if ability in a particular area is greatest at a younger age. The second one is that teams built through bridging relationships make a growing number of the discoveries. As deep knowledge in different fields is enhanced and shared through the local and global teams, partnerships, associates and learning societies, human capital becomes more creative.

Once a creative society is achieved, putting this creativity into the practice, in other words making innovation, becomes the next priority [Yusuf, 2009]. Culture, institutions

and encouragement techniques not only improve creative ability but also facilitate creativity to culminate in innovation. The importance of geographical locus for innovation cannot be incontrovertible besides other aspects. Creativity is likely to lead to innovation in urban cities which are linked to other foreign cities and open to circulation of people and ideas [Yusuf, 2009].

Although the most important ingredient of innovation is human creativity, financial support and commercialization are also significant to making these innovations real. Therefore, innovation can be divided into three partially overlapping processes: the production of scientific and technological knowledge, the translation of knowledge into artifacts, and responding to influencing market demands. As a result, the key to innovation is the active engagement of people and organizations associated with entrepreneurship and risk taking. [Squalli and Wilson, 2014; Pavitt, 2005]

4.2. Project management in innovation projects

Better project management leads innovation projects to success [Tomala and Senechal, 2004]. However, a conventional approach does not cater for practice and theorizing project innovativeness because it cannot deal with nature of innovation projects [Kapsali, 2011]. Therefore, in order to manage an innovation project, it is beneficial to be familiar with the characteristics of innovation projects and its differences from conventional projects. Several aspects need to be considered when comparing conventional and innovation projects [Filippov and Mooi, 2009].

Conventional projects have clear objectives whereas objectives of innovation projects are not necessarily detailed and clear. Innovation is often ambiguous and it is difficult to describe before it is completely achieved. Many innovation projects relate to intangible assets and there can be uncertainty in the commercial success of innovation projects. Innovation is often a result of trial and error. Since objectives are clear and processes are established, risk taking is low in regular projects. On the other hand, risk taking is high in innovation projects considering uncertain tasks and loosely defined objectives. Furthermore, an innovation project team is formed with people from various backgrounds. Therefore, it is expected that separate knowledge residing in individual specialists will be cross fertilized while working in a team.

Conventional project management is shaped by precision, accuracy and optimal use of resources. On the contrary, innovation projects include uncertain tasks and objectives and need slack in resources and time [Filippov and Mooi, 2009]. Therefore, there are significant points that need to be taken into consideration while managing innovation regarding its nature and risks. Furthermore, innovation project management may include different phases than traditional project management for facilitating innovation.

Firstly, managing innovation projects may include more thought processes than traditional project management along with a revolutionary process of facilitating innovation. This way, a project team will be able to come up with innovative results from their project ideas. Additionally, the team will be aware of their mind-set and skill-set and process differences between phases. Therefore, complex innovation management projects can be more manageable with the conscious use of particular mind-sets, skill-sets and activities throughout the innovation lifecycle [Belding, 2014]. The following phases are included in managing innovation project [Belding, 2014].

- Searching – in this phase, ideas which satisfy the needs, respond the opportunities, or foster the organization's strategic goals are investigated and collected. At the end of this phase, opportunities and ideas will be produced which would later become a source of inspiration for beginning of the innovation journey.
- Exploring – ideas and opportunities, which are outputs of the searching phase, are organized, discussed and analyzed in order to comprehend them deeply. It should be ensured that ideas are practical by testing and the proposed innovation needs to attract all of the possible customers.
- Committing – in this phase, the main focus is on actions to be taken, and finding right people committed to support the project manager and team.
- Realizing – this phase concerns achieving goals set in the searching phase.
- Optimizing – attention is given to maximizing benefits; increasing the degree to which the idea has been fully exploited. Optimization or exploitation is considered as central to the concept of innovation.

Apart from the additional phases for innovation projects, project managers need to take the nature of innovation projects into consideration in order to achieve success in their projects. First of all, urgency and innovation are interrelated because companies desire to release new technological products to the market in order to realize profit from the product as soon as possible [Lechler and Grace, 2007]. In such an urgent case, project managers are under pressure. They have to enforce better deadline management and time schedules rise in importance. For innovation projects, project managers should provide flexibility in schedule in order to innovate. Slack time in schedule allows creating and failing and provides strategic flexibility for changes [Lechler and Grace, 2007]. Furthermore, resource allocation with a higher tolerance for slack resources and greater levels of redundancy is instrumental in creating time, space and creativity for innovation [Keegan and Turner, 2002].

Team creativity is a factor that leads to innovation performance. An organization should be structured and arranged to best encourage creativity considering several

factors such as interpersonal dynamics, energy and motivation, openness, leadership, focus, trust, and diversity of skills and experience [Lechler and Grace, 2007]. Therefore, project managers should provide an atmosphere which is able to stimulate creativity of the team, motivate them and make them focus on their tasks.

In order to reduce uncertainty and complexity of innovation projects, risk planning plays an important role [Kapsali, 2011]. Although some risk management could be valuable, excessive or inappropriate risk management might stifle innovation [Vargas and Garcia-Santillan, 2011].

Top management support and project manager authority are the factors which are universally important for all projects. Top management mediates, interprets, buffers the market's needs and allocates the organization's resources. Innovation projects tend to thrive if senior management actively monitors the project. Furthermore, project manager authority and strong management communication are prerequisites for innovation projects. In addition to these, there is a relation between a leader's communicative ability and worker innovativeness. Particular communication techniques can inspire the team's creativity [Lechler and Grace, 2007].

Increasing levels of project management are positively correlated with increasing levels of innovation. However, very high levels of project management are negatively correlated with innovation projects when exceeding a certain limit. Even though project management can facilitate exploitation of existing knowledge, it can also hinder exploitation of new ideas [Filippov and Mooi, 2009].

5. A Case study: anti-patterns encountered in innovation software projects

In order to achieve research questions, a case study has been conducted. In this section, motivation and purpose, and environment of this case study is explained. Furthermore, data collection methodology, collected data, data analysis and results are explained under this section.

5.1. Motivation and purpose of the case study

Software project management is one of the most challenging fields in software development projects [Stamelos, 2010]. Better practice of project management increases the success of software projects. On the contrary, mistakes made in software project management such as resource and schedule mismanagement can result in project failure [Stamelos, 2010].

Software projects are prone to fail due to, in general, the application of constant and wrong solutions continuously to commonly occurring problems [Raptopoulou *et al.*, 2012; Stamelos, 2010; Brown *et al.*, 1998; Brown *et al.*, 2000]. This kind of failures can cause total or partial project collapse (software defects, damaging delays, budget overruns) and give damage to the reputation of software company [Stamelos, 2010]. Software industry is still encountering such failures despite the improvements in development methods and technological advancements [Brown *et al.*, 1998; Raptopoulou *et al.*, 2012].

Anti-patterns are one of the greatest factors which lead to failure in software development projects. There are almost 200 anti-patterns identified thus far [Raptopoulou *et al.*, 2012; Stamelos, 2010; Brown *et al.*, 1998; Brown *et al.*, 2000]. Since anti-pattern structure has recovery solution, understanding anti-patterns provides the knowledge to prevent or recover from them [Jimenez, 2006].

In this case study, the aim is to find out continuously occurred bad project management practices to solve recurring problems in innovation software projects which are more unpredictable, complex and uncertain in comparison with regular software projects. In order to achieve this objective, a survey was prepared for conducting project managers who recently completed to carry out a software innovation project. Questions in the survey are carefully and properly organized as it will be discussed in Section 5.3.

5.2. Case study environment

This case study was conducted to the students from Tampere University of Applied Science (TAMK), Tampere University of Technology (TUT) and University of Tampere (UTA) who got involved in software innovation projects as a project manager carried out by Tampere Demola. Demola associates with these three universities and students of these universities can get credits for their studies and gain experience by taking part in Demola projects which are real life cases licensed by partner companies mostly [Demola, 2015]. Students are able to get involved in a project in their major if there is an available position in a project considering their specialized area. Skills required in a project can include at least one of these; business and concepts, coding, design and art, education, engineering, environment, governance, health care, media and communications and social science [Demola, 2015]. Therefore, it can be said that Demola makes a major contribution to both students and innovation technology in any area.

In this case study, the software innovation projects to be conducted during the academic year 2014-2015 Autumn semester were considered. During the stated time, 33 innovation projects were conducted and one student was assigned to each project as a project manager [Demola, 2015]. Ten project managers responded to the survey among currently ongoing projects.

5.3. Data collection methodology

A questionnaire was designed to investigate project management anti-patterns in software innovation projects. The questionnaire includes multiple choice, scale and open ended questions and consists of two main sections as general information about the manager and examined project and research based questions. In total, there are 29 questions in the questionnaire which can be found in Appendix A. The first part of the first section gathered the information about project manager (age, gender, nationality, and experience in an innovation project as a project manager). The second part includes general questions about examined project such as name of the project, innovation degree in the project, success of the project, applied software life cycle development method and its completeness degree. The third part concerns team size, team experience in an innovation project, disciplines of team members and if there was a problem because of interdisciplinary, its reason and how the project manager dealt with this problem. In the second section, questions regarding followed schedule type during development, tasks on which team spent more time than planned, delay in the schedule and its management, if the project manager followed risk management procedures (plans, control, monitoring and so on), requirements change, source of requirements

change and its management, team creativity and how to engage team creativity into a project are included. At the end, free comments about management of innovation projects are asked to the project managers. The questions were formulated related to possible failures in the management tasks and manager's approach to these failures in order to solve them. While designing the questionnaire, experienced people's opinion from both academy and industry were consulted.

5.4. Data collection

In the case study, 10 project managers from 10 innovation projects answered the questionnaire. The project managers were asked to answer all of the required questions in the questionnaire. The data from the respondents were collected and analyzed considering following sub sections:

5.4.1. General information of project managers, examined projects and teams

First questions of the questionnaire aimed to collect background information regarding project managers, examined projects and teams. Some of these collected data play an important role for the analysis part when combined with research based questions.

Information collected regarding project managers can be found following. Out of 10 project managers who answered the questionnaire, seven of them were female and three of them were male. Age range of project managers can be seen from Figure 3. According to Figure 3, project managers' age ranges between 20 and 30. There were eight different nations as Finnish, Venezuelan, Spanish, German, Italian, Indian, Hungarian and Chinese which project managers were belonged to.

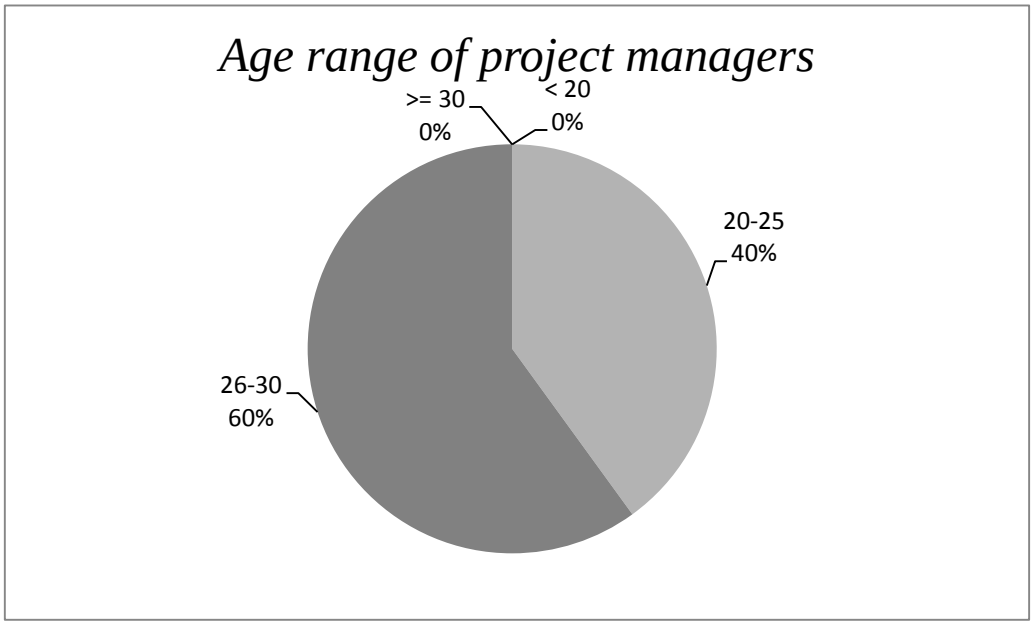

Figure 3. Age range of project managers.

Completed number of innovation projects except the examined one by project managers as a project manager is given in Figure 4. Hence, seven project managers had never been involved in an innovation project as a project manager before. One project manager worked in an innovation project except the examined one. One project manager had completed five projects and another project manager completed eight projects as a project manager.

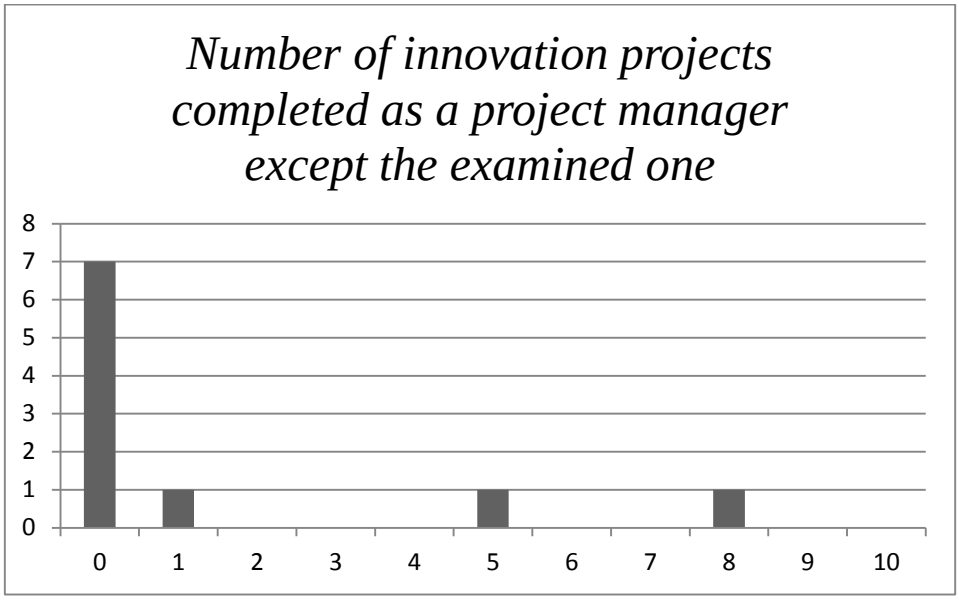

Figure 4. Number of innovation projects completed as a project manager.

Besides project managers' information, it was requested from project managers to give some information regarding the project they were currently working on. The intensity of innovation in the projects can be observed from Figure 5. It can be understood from Figure 5 that one project's innovation intensity was two and two projects' intensity degree was three. Four project managers informed the innovative intensity in the project as four. Lastly, three projects were identified as totally new products.

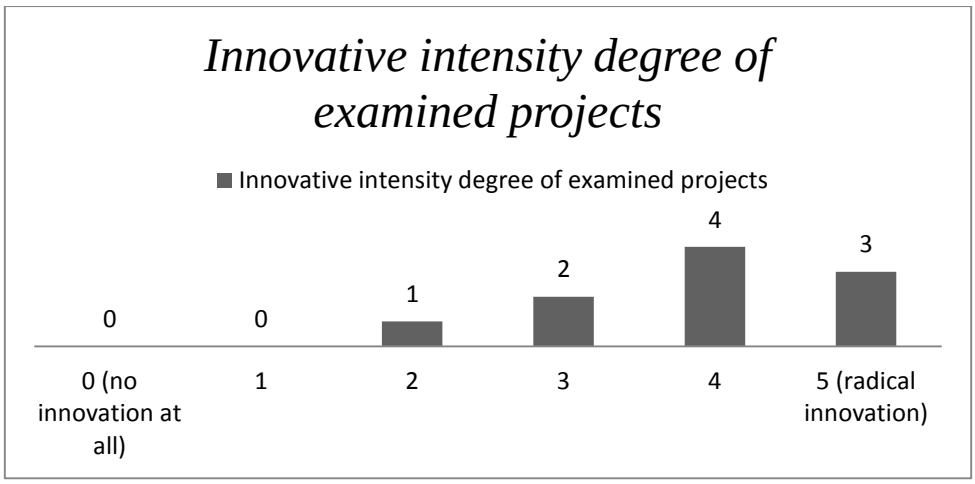

Figure 5. Innovative intensity degree of examined projects.

According to Figure 6, 78% project managers believed that their projects were entirely successful while rest of the project managers thought that their projects were partially successful at the end.

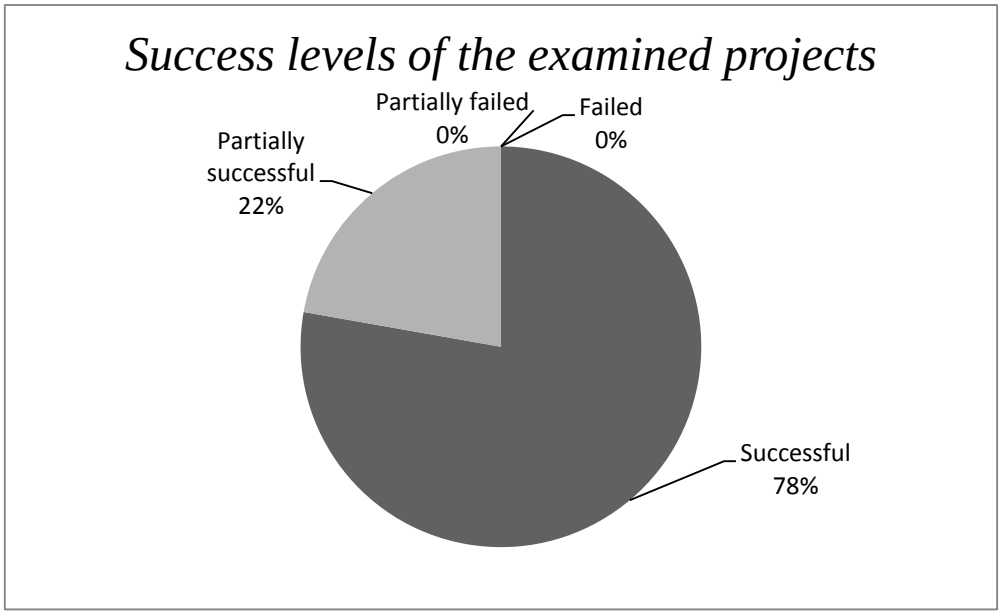

Figure 6. Success levels of the examined projects.

Software development lifecycle models used during the development of the examined projects are presented in Figure 7. According to the associated figure, 50% of the project managers applied agile development model to their projects. While 20% of the project managers did not use any development model, iterative and incremental development model was adapted by 30% of project managers.

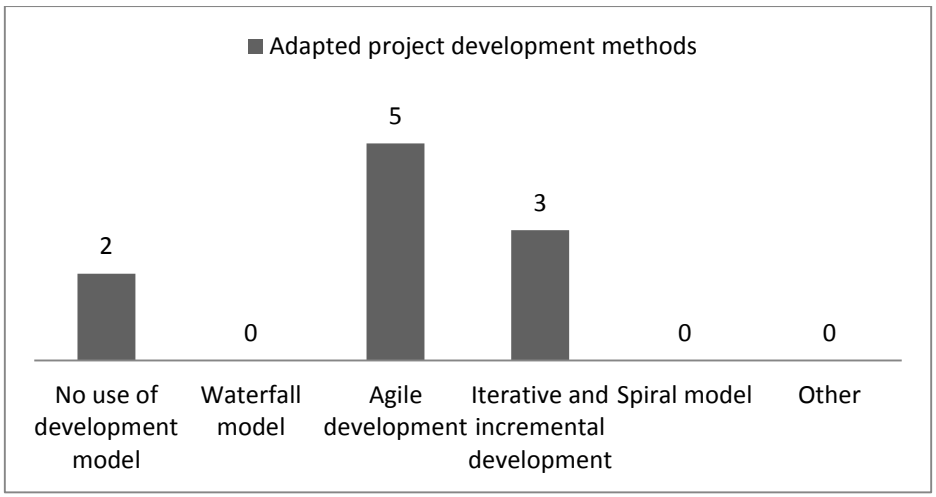

Figure 7. Project development methods used in the examined projects.

Final question regarding examined project concerns completeness degree of the projects when the project completion deadline was over. Completeness degree of projects can be seen from Figure 8. Based on project managers' answers, six projects were able to implement all of the promised requirements. Seventy-five percent of promised requirements were implemented by three projects and one project was able to implement half of the promised requirements.

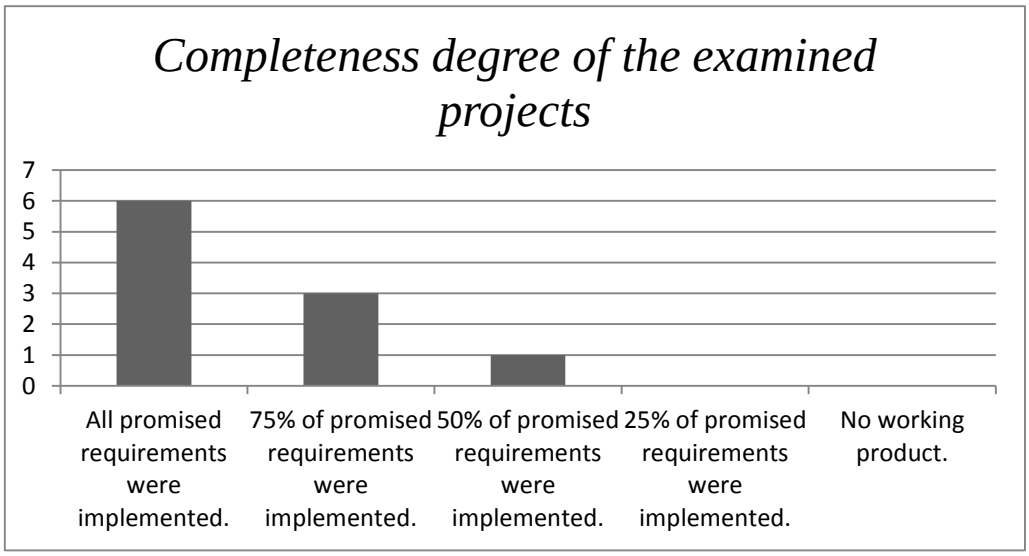

Figure 8. Completeness degree of the examined projects.

Before project managers passed on to research based questions, they were expected to answer questions considering their team which they worked together during development of the examined project. Size of the teams, excluding the project managers, is given in Figure 9. In Figure 9, horizontal axis shows the number of people worked in a team and vertical axis shows number of projects which had associated team size. According to that; one team had two team members, two teams had three team

members. There were three projects with four and five members. One project was implemented by six team members.

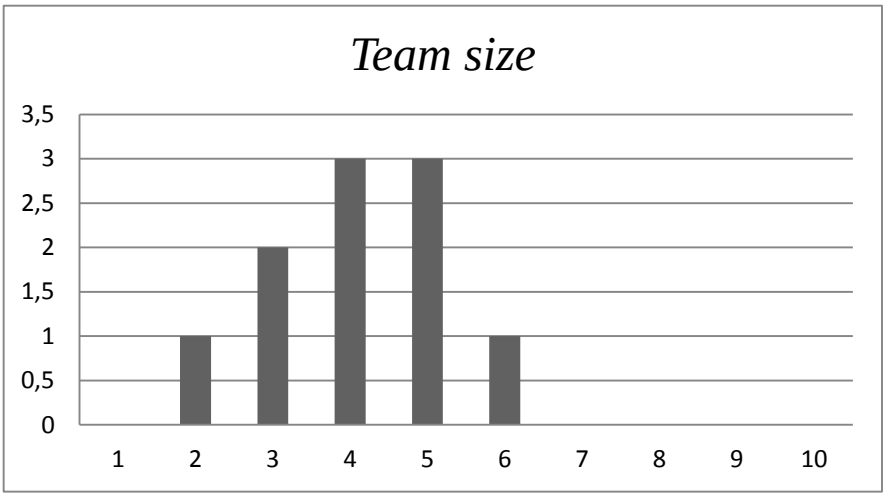

Figure 9. Team size of the projects.

Finally, number of team members who participated in an innovation in the past except the examined one was asked to the project managers. Their answers regarding experience of team members with innovation projects can be seen from Figure 10. According to the bar graph, five projects were implemented by teams which didn't include a member with experience in innovation projects. One project had one team member who participated in an innovation project before. There were four projects implemented by a team including two members who experienced innovation projects previously.

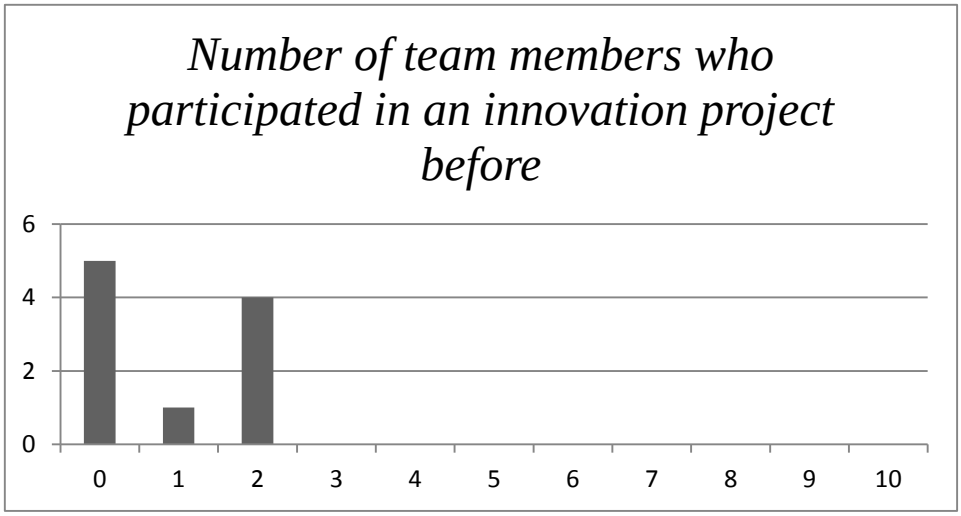

Figure 10. Number of team members who participated in an innovation project before.

5.4.2. Interdisciplinary team

Interdisciplinary team is an integral part of innovation projects. It has been proven that new solutions to complex problems can be developed and research questions can be produced by the efficient use of teams formed with people from various disciplines [Blackwell *et al.*, 2010]. Since the team is formed with people from different knowledge and values, the success of the project will be considered from different angles which may lead to original outcomes. Mistakes made while leading interdisciplinary teams may cause a chaos due to variety of opinions. Even project managers accomplish management of a project successfully; it does not mean that they are great team leaders. Chaos and fighting in the team might be a cause of an anti-pattern such as "Manager not Leader" which points out that a manager is capable of doing administrative and managerial duties but lacks leadership skills [Laplante and Neill, 2006]. In the questionnaire, the question related to interdisciplinary teams consists of two sub-questions. It was asked to the project managers if their team included members from different backgrounds as a main question. As it can be seen from Figure 11, almost all of the projects were implemented by an interdisciplinary team except two according to the project managers.

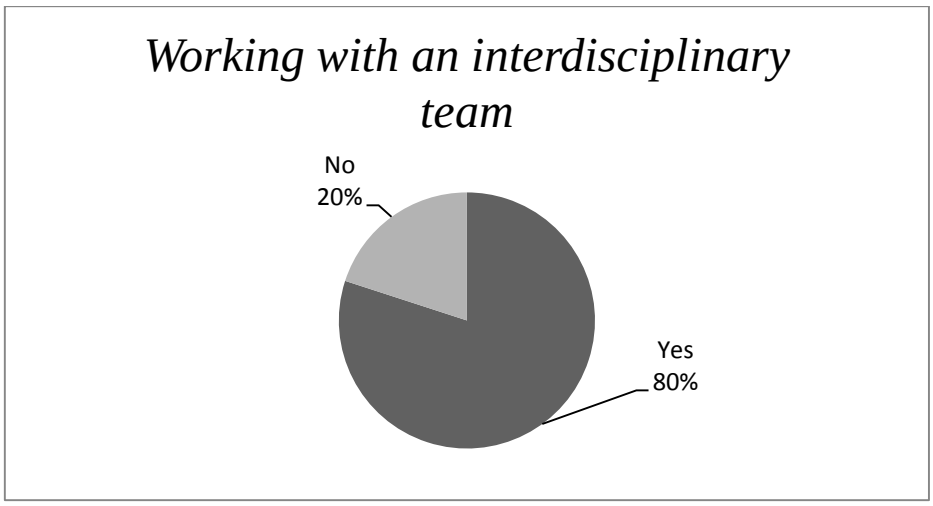

Figure 11. Number of interdisciplinary teams in the projects.

Different disciplines which worked together in the examined projects can be seen from Table 3. All project managers answered the question if there were any difficulties because of the interdisciplinary team and how they managed the situation. Seventy percent of the projects did not have any difficulties regarding interdisciplinary. However, 30% percent of the project managers had some problems in managing their interdisciplinary team. Results for this question can be observed from Figure 12. Twenty percent of the projects had difficulties because team members who were from different disciplines didn't understand other disciplines work. According to a project

manager's words, "a*rt guys didn't understand software side. So they sometimes delivered graphics that just couldn't work together with the software*" and another project manager complained "*sometimes it was hard to explain to the designers what was easy to implement and what was not*". In brief, designers made suggestions to the software developers to implement infeasible features for the product even they did not want to understand that it was impossible to implement their ideas. The reason of the difficulty in another project was the lack of team members in a specific field. Therefore, the team lost so much time in trying to learn unfamiliar subjects to their fields. From Table 4, the actions of project managers who did not have any problems with the interdisciplinary team in order to prevent any possible problems caused by an interdisciplinary team can be observed. Moreover, project managers who had difficult times with the interdisciplinary team shared their way to recover from this problem. Their answers can be seen from Table 5 as well.

Project	Disciplines
1	Security engineering, graphic design, interactive media and software development
2	Language, literature and translation, information studies and interactive media, human-technology interaction, web development
3	Scriptwriting, graphic design, business, interactive media and media management
4	Educational sciences, business, interactive media, language, literature and translation
5	Business, graphic design, interactive media
6	Business, interactive media, logistics, information sciences
7	User interface design, computer science, videogame design, business
8	Civil engineering, environmental engineering, electrical engineering, computer science, biomedical engineering

Table 3. Disciplines in cooperation in the examined projects.

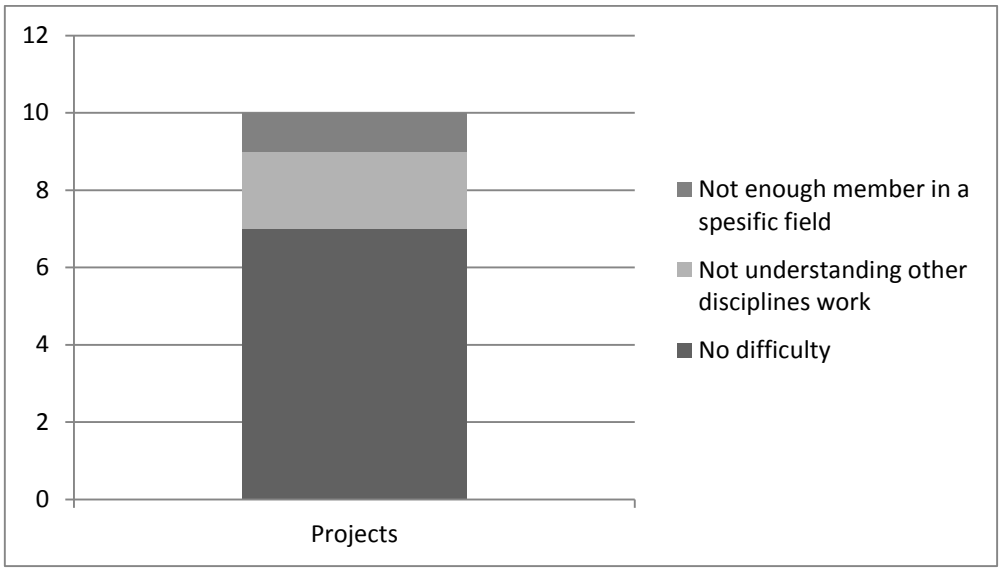

Figure 12. Difficulties because of the interdisciplinary team.

Actions to manage interdisciplinary teams
Communication with the team
Helping other team members from different disciplines
Positive attitude
Being open
Giving tasks to the team instead of assigning individual tasks
Dividing the team as same disciplines work in the same sub-teams
Increasing team motivation
Understanding human psychology

Table 4. Actions to manage interdisciplinary teams to prevent from possible problems.

Enforced actions by project managers to recover from interdisciplinary problems
Making the team help each other to understand a particular discipline better
Having communication with the members who made the process harder
Doing team members' job and fixing problem parts in the project

Table 5. Applied actions by project managers to recover from interdisciplinary problems.

Besides problems arising because of interdisciplinary teams, it was also asked to the project managers if different backgrounds (cultural, national, organizational…) of team members caused any serious problems. Figure 13 shows the percentage of the teams which had problems because of different backgrounds of the team members. The reasons behind the problems were reported by project managers as team members with different disciplines, inexperienced team members, different interpretation of work ethics and cultural differences in brief. Two project managers lived through similar problems and explained their problems as "*we had a conflict in the way that one person was less committed to our project and that made work effort unequal in the team*" and "*one of our developers was not committed to the project, had no experience or English skills to work with*". Additionally, a project manager explained the situation about a team member who did not share the same objectives with the team as "*money was the only issue. One team member left the group since his goals were based mainly on moneymaking and others were not able to keep up with him*". Furthermore, another project manager expressed the problem arising because of cultural differences with these words; "*the guy from design and media was from Portugal. Cultural differences came up several times during the project. For him, being late was acceptable and he seemed not to understand why we considered it a huge problem*".

As a solution, one project manager suspended inexperienced team member from the project. In addition, in another project, a team member, who had different work ethics than others, did not share the same purpose with the rest of the team and worked only for moneymaking purpose left the team. These problems caused unbalanced work division within the team and slowing down the progress of the project.

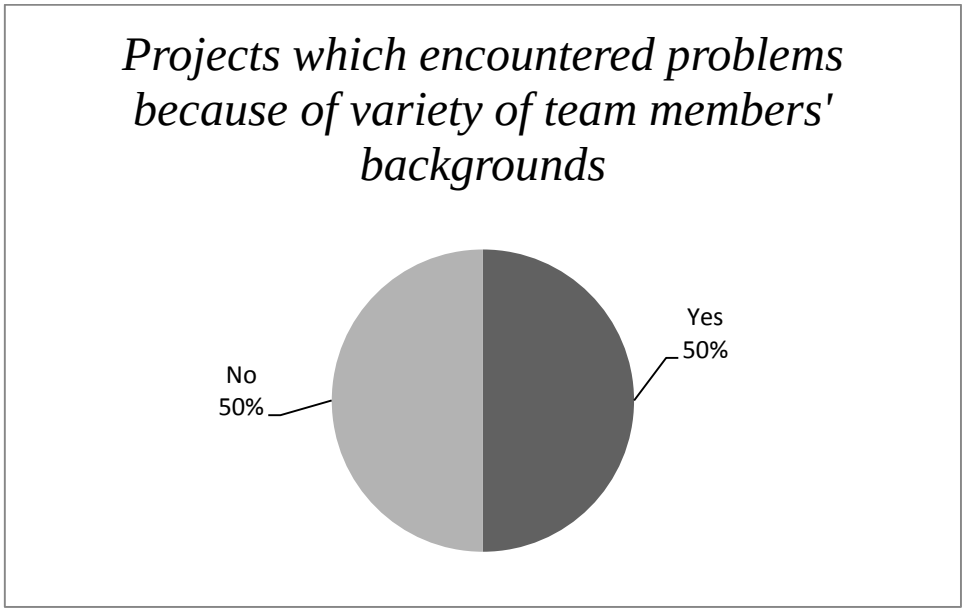

Figure 13. Projects which encountered problems because of variety of team members' backgrounds.

5.4.3. Controlling the schedule and phases

Proper management of the project schedule and phases defined in this schedule thrive the project. Deadlines of deliverables and tasks should be determined wisely considering a project's complexity, size, and team capabilities. Weakly designed schedules may cause delays. Additionally, spending more time in a phase than planned makes difficult to move on next phases. Motivation of the team members can fall, delays in the schedule can put stakeholders in stress, poor quality works can occur to make up for the lost time, even cancellation of the project can be considered because of poor schedule management. This situation is more serious for innovation projects since they are high risk cases. Therefore, providing flexibility in the schedule becomes a must for innovation projects. There are examples of anti-patterns in the literature regarding this situation. For example, "Death by Planning" and "Analysis Paralysis" concern failure and delay in phases, respectively, planning and analysis. In order to find out schedule related problems encountered by project managers, five questions were asked to the project managers in the questionnaire.

According to the questionnaire results, 20% of project managers followed a carefully planned and firm schedule whereas 80% of projects' schedule was flexible. Beside of this, 60% of project managers spent more time with task(s) than they planned. Forty percent of project managers completed the tasks on time as they were scheduled. Tasks which have required more time than planned in the projects are presented in Figure 14. According to Figure 14, planning was the phase in which teams spent more

time than the project managers planned. A project manager specifically explained why they were late in planning phase and other tasks "*because ideas and creativity cannot be forced so it was not possible to keep deadlines if a good idea only came afterwards a second reason was a not so engaged working attitude of some of my team members that made us take more time for a task than planned initially*". Seventy percent of projects experienced delay and the reasons behind the delay in phases were explained as it can be seen from Table 6.

Seventy percent of the project managers amongst project managers who experienced delay were able to manage the delay in the task(s) while 30% of the projects encountered serious problems because of delays. Three of the project managers, who had delay in the tasks because of an irresponsible team member, defined the way how they managed the situation as "*I tried to ask for support from the other team members, to do more work so we can still keep our progress up and going. I also put more work myself into the project. And we had open discussion about the problems within the group*", "*we did his job and we set inner deadlines than the real deadlines*" and "*one person was taken out of the team and replaced by a committed skilled developer*". Among the projects which had serious problems because of delay in the tasks, one project was late and not able to finish the project before the deadline and the other one let a team member leave the project and continued one member missing to the project which had a severe effect on other team members' motivation.

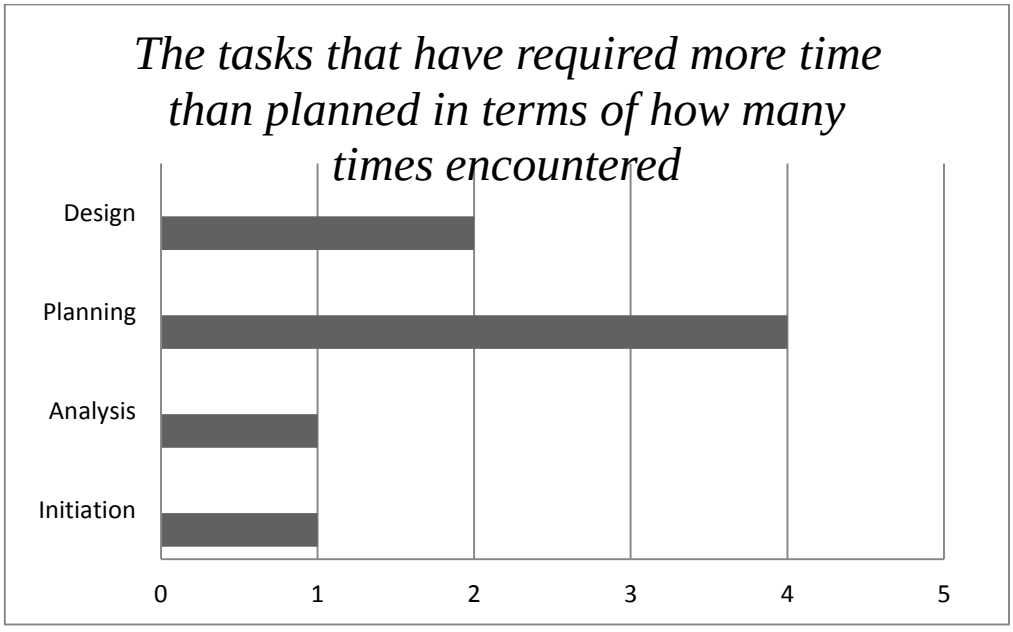

Figure 14. Delayed tasks in terms of how many times encountered.

Project	The reason behind the delay in the tasks
1	Overwhelming assigned tasks, insufficient number of coder, other studies that should have been paid attention besides this project, no motivation
2	Inexperience in the software and the required language to implement the project
3	Inexperienced developers
4	Insufficient guidance of the project partner, Christmas break, elapsed time while producing ideas
5	Elapsed time while producing ideas, not committed team members
6	Unmotivated team members
7	Irresponsibility of a team member, other studies that should have been paid attention besides this project

Table 6. Reasons behind delay in phases.

5.4.4. Risk management

Innovation projects are known more risky than regular projects because these projects may involve unpredictable changes, uncertain tasks and unclear objectives. Effective risk management strategies contribute to the project success by preventing potential problems and providing opportunities to the project. Therefore, more emphasis should be given to risk management especially in innovation projects. While lack of risk management can be a cause for an anti-pattern, its effective application to the project can be a refactored solution itself. For example, typical causes of "Project Mismanagement" anti-pattern include lack of risk management [Shvets *et al.*, 2015]. The refactored solution is proper risk management which can resolve predictable symptoms and consequences of "Project Mismanagement". In order to understand if project managers applied effective risk management to their projects, three questions were asked to them in the questionnaire. According to Figure 15, 40% of project managers prepared proper plans (e.g. specifications, architecture, and risk management plans) and it helped a lot during development of the project. Thirty percent of projects' plans were not sufficient while there was no plan at all in the case of 20% of projects.

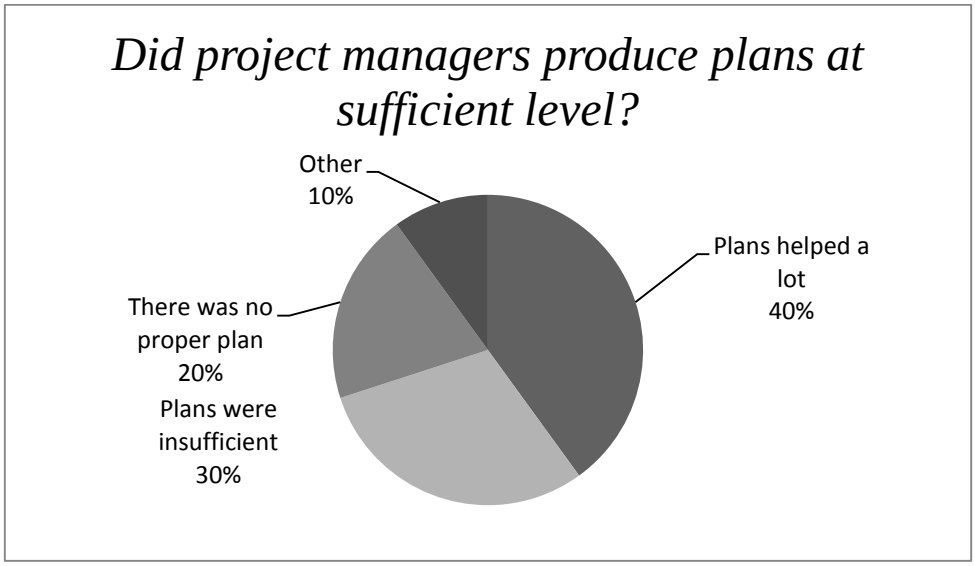

Figure 15. Whether project managers produced plans at sufficient level or not.

In Figure 16, specific control and monitoring techniques applied by project managers can be seen. According to Figure 16, although most of the project managers with 40% ratio applied "Other" methods as control and monitoring techniques, none of them specified what they were. Thirty percent of the project managers were not able to control and monitor the project. Twenty percent of the project managers performed random code reviews and minimal software inspections whereas 10% of the project managers evaluated code reviews and software inspections frequently.

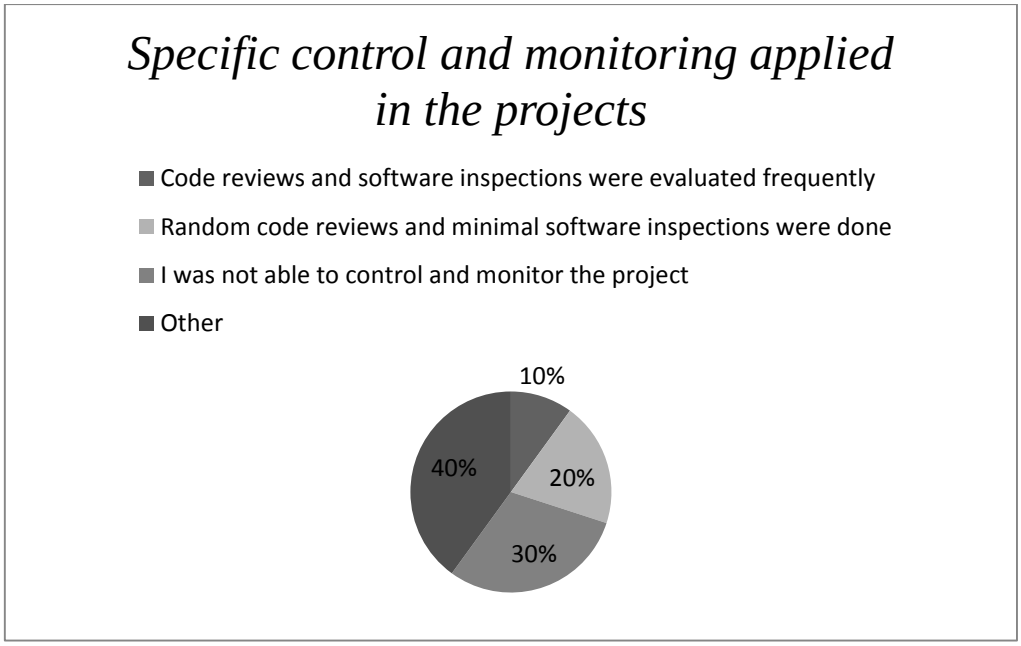

Figure 16. Specific control and monitoring applied in the projects.

Inadequate planning, control and monitor of the projects caused skipping test phase in one project. According to this project's manager "*Time management was a problem. There should be clear plan and schedule for the whole team. We were not able to do this, since it was an innovative project, we used the time we thought was we needed. Meetings were set one week in advance, about 2-3 time per month. We had no testing phase since we ran out of time*". Furthermore, one project manager mentioned difficulty of producing plans in innovation projects with these words; "*for innovation project, at first the requirement is totally unclear so it is hard to have a plan to develop the final product. So it cost us much time to come up with ideas*". One project manager made a self-criticism "*If I could change something now, I would do a more detailed plan for the whole project instead of having occasional weekly plans. We had a very good plan at the beginning but we lost our touch during the middle*". There were no problems because of inadequate planning, control and monitoring for the other projects which did not have a proper plan, control and monitoring as their managers did not mention about any problems.

5.4.5. Requirement change management

Requirement change in innovation projects is an inevitable fact. Using effective and efficient change control techniques helps to achieve project objectives. Conversely, poor change management is one of the biggest reasons of project failure. Wrong techniques applied as a part of change management can be a cause for anti-patterns. For instance, "Change Prevention Process" anti-pattern concerns proper change management process. If a project manager ignores new requirements and prevents them to be added to the project, "Change Prevention Process" anti-pattern can occur. As a result, stakeholders will not be able to get what they actually need [Ambler, 2015].

Firstly, it was asked to the project managers if there was any changes in project requirements and how frequent it was. Answers of the project managers can be observed from Figure 17. Thus, 10% of project managers mentioned that their project requirements always changed, 30% of the projects' requirements changed occasionally, other 30% of the projects' requirements changed often and it was told by remaining 30% of the project managers that project requirements changed rarely.

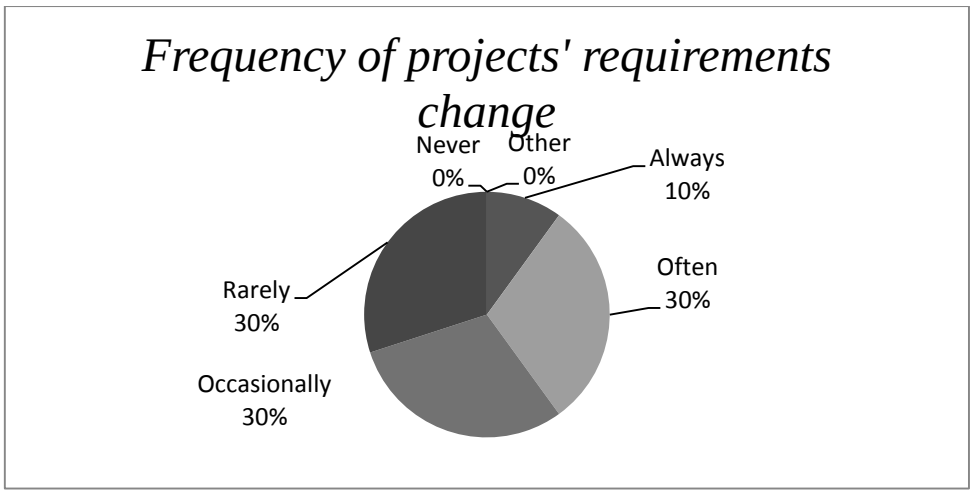

Figure 17. Frequency of projects' requirements change.

Main sources of requirement change in the projects varied as it can be observed from Figure 18. According to the project managers who answered the questionnaire, the main source of changes was team members' ideas and clients' demands and requirement change because of technical reasons was observed in one instance.

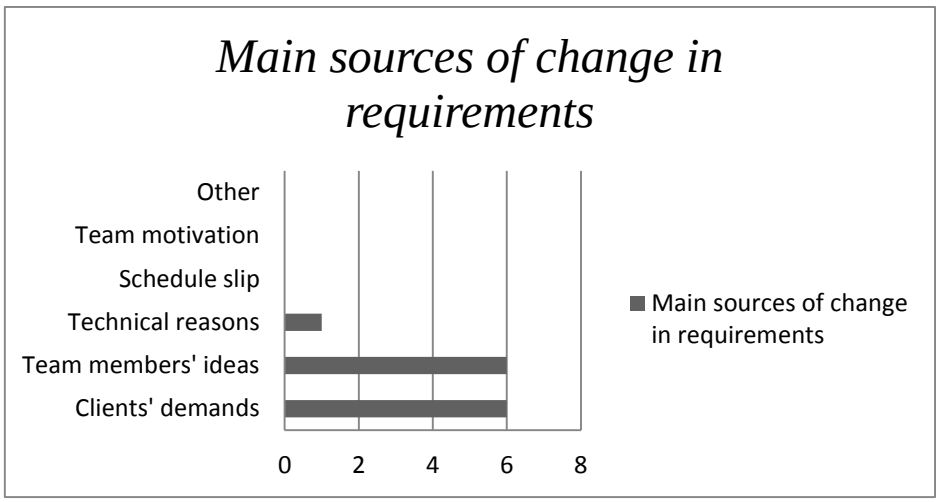

Figure 18. Main sources of change in requirements.

According to the project managers, they were able to handle changes in requirements and they did not have any trouble with requirements change. Most of the project managers followed the similar ways while managing requirements change. They emphasized the importance of communication with the team members and the clients whenever there was a change considered as they mentioned in the questionnaire; "*Quickly give out tasks to the team members to prepare changes. Plan team meetings to work on it. Communicate the urgency to every team member.*", "*By discussing the changes with the team first to achieve a common understanding for the context, then with the client to get their perspectives and again following up with what could be done and how the changes could be implemented.*" and "*when the changes were considerable*

we always tried to negotiate to the client to reach a feasible solution/change". For one project, requirements change changed the almost all project objectives and roles assigned to the coder. According to the project manager "*it took some time to understand what the new role for our coder actually means and how that coder's knowledge can be utilized for the tool's benefit*". In another case, lack of customer involvement while shaping the product caused producing infeasible ideas by team members as the project manager explained the situation "*because our customer didn't give us any idea, so every time we show our ideas to the customers we would change some and some ideas were not easy to implement*". In one project, requirements change was manageable with extra work of the team members.

5.4.6. Making use of team creativity and its engagement to the project

Creativity and innovation are interrelated terms. Before developing something new, there should be an idea of it. Creativity produces innovative ideas and innovation makes use of these ideas. Triggering, maintaining and developing team creativity leads to innovation performance. In the questionnaire, it was asked to the project managers if team creativity was undermined at some point and how to engage team creativity into innovation projects.

From Figure 19, factors which undermined team creativity in the projects can be seen. According to this figure, 34% of the projects' team creativity has never been undermined. However, remaining projects' team creativity were undermined because of working environment, tight schedule and too much customer involvement as their ratio can be seen from Figure 19.

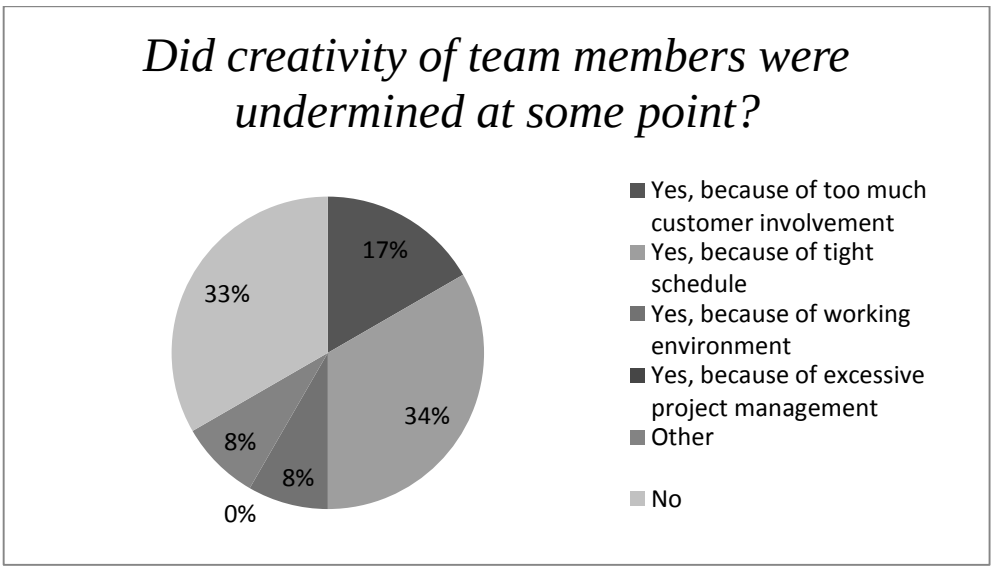

Figure 19. Factors which undermined team creativity if team creativity were undermined.

Furthermore, it was asked to the project managers to share their own ideas and experiences about how to engage team creativity as a project manager into the innovation projects. Their answers can be found below.

One of the project managers made emphasis on the importance of open communication and motivation of the team in order to engage team creativity; "*open communication and motivation of the team. Plan regular meeting to keep everyone involved, as well as give out regular work tasks to keep the workflow going. In addition, I also showed my own engagement by giving as many ideas and work input myself, to motivate the other team members.*"

Besides communication, most of the project managers believe that engagement of team creativity requires brainstorming and sharing new ideas among team members. They explained their thoughts on this issue with following words; "*at first, when we had to start from scratch, I asked everybody to come up with some ideas of how our project data could be used innovatively in teaching. These ideas were then presented in our weekly meeting, merged and together developed further. I am also not the most creative person, but I like to ask a lot of questions, so at first I mostly just let the other team members to come up with new ideas and scenarios and then ask questions of how could those ideas be actually implemented in our tool.*" and "*in the beginning we met quite often and tried to get to know each other. Talking and just brainstorming were the best ways to create something new. We kept on just brainstorming until we had every crazy idea on paper and the next time we started all over again, since we felt that the thing done last time only partially worth developing. We made decisions together and everyone got to take part in and share ideas.*", "*we did many brainstorming sessions in the beginning and tried to meet frequently. I was constantly discussing ideas and changed with the designers and backend developer during implementation.*", "*I think it is very important that team members have a lot of creativity and new ideas and that they share them with others because it helps a lot to develop the project*", "*in our meetings we had very casual and open discussion about what kind of improvements there should be to our project*" and "*when you think of ideas, you should accept others ideas. Combine them and make them clear to show, then you have to discuss with your team members that which ideas are good to choose*".

One of the project managers suggested giving every team member the freedom in a planned and controlled manner in order to engage team creativity in innovation projects. Another project manager recommended letting team members see the limits of the project and what they can accomplish as a team. By this way, team members can see the "*big picture*" and get motivated. One of the project managers made a suggestion on how to increase motivation of the team members by informing them about positive

feedbacks of the project partner; "*I was trying to inform everything about the opinion of the project partner, so I shared the positive feedbacks with them from our mailings, so they could be proud also. But in our case, I had an easy task, because most of us were motivated and hard-working*".

5.5. Data analysis and results

Collected data from project managers is analyzed considering mostly encountered anti-patterns available in the literature. Symptoms and consequences of anti-patterns and the way project managers tried to solve problems are analyzed carefully and compared with the collected data. After identification of anti-patterns encountered in the examined innovation projects, it is reviewed whether a new anti-pattern which is particular to the management of innovation projects can be interpreted from the collected data. In this section, software project management anti-patterns found in each examined innovation projects will be presented and a possible new anti-pattern specific to innovation projects will be described.

5.5.1. Anti-patterns observed in Project 1

Anti-patterns encountered in Project 1 are "Project Mismanagement", "Architects Do Not Code", and "Micromanagement".

The project manager did not employ any specific control and monitoring and there was not a kind of plan at all. "Project Mismanagement" anti-pattern concerns control and monitoring of the projects. Lack of risk management activities can cause problems in the projects and "Project Mismanagement" is observed in such cases. Therefore, Project 1 encountered "Project Mismanagement" anti-pattern.

Another anti-pattern observed in the associated project is "Architects Do Not Code". This anti-pattern occurs when software architects are not included in the implementation phase. For this reason, they are not interested in implementation details and can deliver flawed design. In this project, designers delivered graphics which could not work with the software. Since they were not involved in the implementation phase, these flawed graphics were fixed by the project manager. Therefore, "Architects Do Not Code" anti-pattern occurred in Project 1.

As it is mentioned above, project manager involved in the implementation phase and fixed the flawed graphics. This behavior of the project manager conflicts with the software project management responsibilities. Programming requires full attention of developers. Project management requires full attention of project managers in the same manner. "Micromanagement" occurs when project manager changes his/her task priorities. In Project 1, project manager did the most of the coding. Therefore,

“Micromanagement” is observed in this project. When project managers start to do developers’ responsibilities, team members can feel less competent and their talent level can be lowered. Since project managers do not show required attention to the managerial tasks, management help will be needed.

5.5.2. Anti-patterns observed in Project 2

There is no anti-pattern observed in Project 2 with the available data. The project was successfully completed without any conflicts between team members and problems. The project manager was one of the experienced project managers among others. The project manager participated in five innovation projects as a project manager previously. Success of Project 2 can be associated with the project manager’s experience as the project manager mentioned in one of the questions. Furthermore, the project manager associated his success to the importance of understanding human psychology in such projects which involve management of people.

5.5.3. Anti-patterns observed in Project 3

Software project management anti-patterns observed in Project 3 can be listed as: “Project Mismanagement”, “Corncob”, “Leader not Manager”, and “Micromanagement”.

Since there was no control and monitoring of the project, “Project Mismanagement” is observed in Project 3.

“Corncob” is observed in Project 3 because of one team member’s irresponsibility. This team member’s behavior caused delays and unbalanced work division between team members.

Although the project manager dealt with “Corncob” situation and it did not cause any problems at the end, the project manager was not able to include other responsibilities of the team members to the project plan since they were all students. This situation caused missed targets so that delays in planning, design and closing phases and unfocused team members. Exam dates are generally listed in the academic curriculum at the beginning of each academic year and teachers explain the responsibilities of students and their possible assignment dates during introductory lesson of a particular course. The project manager should have collected this information from team members and included them in the project plan. Therefore, “Leader not Manager” anti-pattern is observed in Project 3.

As in Project 1, the project manager of Project 3 did coding and other developers' job to make up the lost time. Hence, "Micromanagement" is another observed anti-pattern in Project 3.

5.5.4. Anti-patterns observed in Project 4

"Project Mismanagement" might be observed in this project because of minimal software inspection and random code reviews.

5.5.5. Anti-patterns observed in Project 5

"Project Mismanagement" and "Manager not Leader" anti-patterns are observed in Project 5.

The project manager did not employ and control and monitoring of the project. Thus, "Project Mismanagement" anti-pattern is observed in this project.

Apart from "Project Mismanagement", another anti-pattern observed in this project is "Manager not Leader". The project manager was the most experienced one among others with eight innovation projects completed as a project manager previously. The project manager was already aware of the managerial duties and had a chance to experience them several times. Although a project manager should accomplish leadership and managerial duties, it does not mean that one can be a leader and a manager at the same time since they require different capabilities. The project manager did not have any problems in managerial duties according to his/her answers. However, the project could not be completed because of low motivation of team members. Leaders are responsible from keeping team members' motivation high. Hence, the project manager was not a leader but a manager. Therefore, "Manager not Leader" is observed in Project 5.

5.5.6. Anti-patterns observed in Project 6

"Project Mismanagement" is observed in Project 6 because minimal software inspections and random code reviews were done.

5.5.7. Anti-patterns observed in Project 7

Project 7 encountered "Project Mismanagement", "Corncob", "Architects Do Not Code", "Micromanagement" and "Size Isn't Everything".

Since there was no control and monitoring of the project, "Project Mismanagement" anti-pattern is observed in Project 7.

The project team had difficult times with team members who were not committed to the project. As a solution, one of the members was replaced with another committed team member. Therefore, "Corncob" is observed in this project.

"Architects Do Not Code" anti-pattern is also observed in this project. Like in the case of Project 1, designers delivered graphics which were not implementable to the project.

As similar to Project 1 and Project 3, project manager took part in implementation part. Thus, "Micromanagement" is observed in Project 7.

In one question, the project manager said "*less is more*" about the team size. The team started to the project with five members. However, because of problems caused by two members, the project manager claimed that they would have been more successful if they were three at the beginning instead of five. Sometimes less number of people can accomplish more achievements. In such cases, "Size Isn't Everything" anti-pattern occurs in this project.

5.5.8. Anti-patterns observed in Project 8

Anti-patterns observed in Project 8 are "Corncob", "Project Mismanagement", "Road to Nowhere" [Laplante and Neill, 2006], and "Untested but Finished".

In this project, a team member did not share the same objectives with other team members. Therefore, the team member was not productive through the project development process and left the project. The team member's choices affected the project negatively. Other team members had to put more effort to the project and it had a severe effect on their motivation. Therefore, "Corncob" anti-pattern is observed in this project.

Since there was no control and monitoring of the project and insufficient plans were produced, "Project Mismanagement" anti-pattern is observed.

The project did not have a proper and clear project plan. Thus, time management problems occurred and the project started to use time when they needed without a plan. Therefore, "Road to Nowhere" anti-pattern is observed in this project. Without a proper and updated plan, the project progress cannot be tracked.

Additionally, since the project manager did not prepare a clear project plan, they were not able to use the time efficiently. They had to skip testing phase because of lack of time. When projects run out of time, project managers are tend to skip testing phases. An untested software product can have security gaps and bugs which reduce its quality. For this reason, the project experienced "Untested but Finished" anti-pattern.

5.5.9. Anti-patterns observed in Project 9

"Glass Case Plan", "Gilding the Lily", "Train the Trainer", and "Leader not Manager" are the anti-patterns observed in Project 9.

The project had a clear plan at the beginning of the project. However, the project manager could not maintain this plan and update it. Hence, team members did not know their next tasks until the project manager told them. "Death by Planning" anti-pattern occurs in two cases: "Glass Case Plan" and "Detailitis Plan". "Glass Case Plan" occurs when there is deficient and outdated plan. Therefore, the project encountered "Glass Case Plan" anti-pattern.

"Train the Trainer" is another anti-pattern observed in Project 9. When a team does not have any background knowledge about the project, it is sometimes easier to let one member to learn the subject and teach rest of the team instead of trying to learn main subject of the project as a whole team. By that way, project teams can save time. Project 9 was about the education system of a particular country and team members did not have any idea in this subject except one of them. Even though they had a member whose background was education, they spent too much time with understanding the education system (including visiting a school) as a team. According to the project manager, because of the time they spent while learning the education system they were not able to start the project on time. Therefore, Project 9 experienced "Train the Trainer" anti-pattern.

"Gilding the Lily" anti-pattern occurs during requirements analysis and design phase where requirements are likely to change. Project manager cannot handle the change and it causes delayed development. Design and implementation issues can be reintroduced in analysis which leads analysis phase to exceed. In this project, the project almost completely changed with the changes in requirements in the middle of schedule. For the project manager, it took so much time to understand the new requirements of the project and changing role of the developer. Therefore, task assignments were done late. As a result, "Gilding the Lily" is observed in this project.

5.5.10. Anti-patterns observed in Project 10

"Corncob" anti-pattern is observed in Project 10 because of a not committed team member.

As a consequence, summary of observed anti-patterns in the examined projects is presented in Table 7. According to Table 7, the most encountered anti-patterns in students' innovation projects are "Project Mismanagement" with seven times

observation, “Corncob” observed four times, “Micromanagement” observed three times, and “Architects do not Code” observed two times.

Projects	Anti-patterns
1	Project mismanagement, Architects do not code, Micromanagement
2	No anti-pattern
3	Corncob, Leader not manager, Micromanagement, Project mismanagement
4	Project mismanagement
5	Manager not leader, Project mismanagement
6	Project mismanagement
7	Corncob, Architects do not code, Project mismanagement, Size isn’t everything, Micromanagement
8	Corncob, Project mismanagement, Road to Nowhere, Untested but finished
9	Glass case plan, Gilding the lily, Train the trainer, Leader not manager
10	Corncob

Table 7. Summary of anti-patterns observed in the examined projects.

Apart from the anti-patterns which are already available in the literature, a possible new anti-pattern is discovered in this research as it is described in Table 8. Creativity is an integral part for innovation projects. The idea was explained in detail in the Section 4. It was asked to the project managers if team creativity has been undermined at some point during development of the project. Their answers to this question can be observed from Section 5.4.6 and Figure 19. Furthermore, project managers shared their views regarding how to engage team creativity into the project as it can be observed from Appendix B. As a result of their answers, team creativity can be undermined by too much customer involvement, tight schedule and working environment. However, it is an important aspect to manage team creativity and bring balance to the reasons which hinder team creativity. In Table 8, name, central concept, generic and unique causes, unbalanced forces, symptoms, consequences, identification and refactored solution of this new anti-pattern are explained.

Name	Noncreative intelligence
Central concept	A project group is formed and an innovative project idea is assigned to this group. They need to improve the idea assigned to them by producing enough number of creative requirements. However, sometimes it is not easy to come up with creative ideas because of particular reasons. The team spends too much time to produce creative ideas in the analysis and planning phase. The project manager cannot accelerate this process since team creativity cannot be forced or creativity is a gift. Therefore, they have delays in the schedule and analysis and planning phases can extend. Delays in initial phases prevent further process in software development lifecycle.
Generic and unique causes	Lack of management's ability to provide a working environment which stimulates team creativity. Too much customer involvement which desire imitated requirements in the product and prevents team creativity. Tight schedule.
Unbalanced forces	Management's apathy and unawareness.
Symptoms	Projects get stuck in planning phase. Innovative ideas should be produced and analyzed in the first stages of a software development process. Teams gather in meetings to come up with innovative ideas and achieve no useful result.
Consequences	Motivation and team confidence lost. Delays in analysis and planning phases can occur.
Identification	The project manager expects to produce some innovative ideas in next meetings. Team anxiety arises.
Refactored solution	Project manager should include a phase regarding stimulation of team creativity, revealing creative ideas, and making decision among those ideas. Additionally, project managers should find out the causes which hinder team creativity (i.e. tight schedule, too much customer involvement, working environment and so on) and produce solutions according to the cause. For instance, if team creativity is undermined because of too much customer involvement, the project manager should bring balance to the relationship with the customer. If the reason is tight schedule, slack time should be included into the schedule. If the working environment is the problem, the project manager should provide a working environment which creates an atmosphere to stimulate team creativity.

Table 8. Description of the new anti-pattern.

6. Conclusions, limitations and future work

Software projects are prone to failure because of their complex and perplex nature. Software innovation projects have more risks than conventional projects. This fact alone makes it more difficult to manage innovation projects, since the following might occur:

i) project objectives may be unclear and unsettled for most part of project lifecycle;
ii) activities and tasks may not be adequately defined or well-defined;
iii) changes may be unpredictable, and thus uncontrolled.

One of the reasons behind software project failure(s) is the continuous application of rather wrong solutions to a particular problem which occurs over and over again. In other words, anti-patterns are one of the greatest failure factors of software projects.

Project management anti-patterns concern mistakes made in software development processes and define the key scenarios which give damage to software processes [Brown, 1998]. A project manager can study anti-patterns and be aware of commonly occurred problems, attempted bad practices to solve the encountered problems, and refactored solutions. Therefore, the professional project manager can prevent problems or recover from the problems. It can be claimed in a nutshell that a project manager with anti-pattern knowledge and expertise will bring extra value to the project and likelihood for potential success.

In this study, software project management anti-patterns encountered in innovation projects have been observed, identified and analysed. A case study, in order to achieve the research objectives, was conducted among advanced MSc students, who were project managers in innovation projects in industry in the autumn term of 2013-2014 academic year. All the examined projects were real life projects assigned by Demola itself or Demola's project partners [Demola, 2015]. A questionnaire was disseminated to the student project managers in order to reveal the problems they experienced through and their management approach to solve these problems. Ten project managers out of 33 project managers responded to the questionnaire. The collected data was analyzed carefully and subsequently was compared and contrasted to the already existing anti-patterns as documented in the project management literature.

Following the research findings, it can be stated that anti-patterns encountered in innovation projects show similarity with the anti-patterns in conventional projects. However, a new and different anti-pattern regarding team creativity was discovered, and this is as another revealing result of this study. Since creativity is more involved in innovation projects than regular projects and since creativity and innovation are,

actually, interrelated terms, the following can also be stated as one of the research outcomes: An anti-pattern which makes emphasis on the importance of team creativity and its stimulation has been found in innovation projects. Therefore, what is important for project managers is that team creativity should be managed and handled in innovation projects.

The study fills in a gap in anti-pattern research since it is the first ever anti-pattern research study conducted considering anti-patterns found in innovation projects. At least, to the current knowledge available there has been no prior related research study for innovation projects' anti-patterns although there are some available anti-pattern research studies for regular projects. Hence, the first limitation of this research has been experienced during the literature review stage because of lack of prior research findings for reference and comparison and for recommended research methodology in the field. The second difficulty and limitation of this research study was that it was not easy to collect answers from the respondents to the questionnaire. Therefore, too much time than expected was spent on collecting responses from an admittedly limited number of project managers. However, this collected data analysis led to the first ever empirical knowledge available on innovation projects' anti-patterns, and although indicative, it still constitutes a starting point for further research in anti-patterns and their effective management.

Anti-patterns are an open-ended topic that should continuously be observed in order to get accurate and consistent results. There will always be room for more research and development work. It is possible to carry out another research study in the future which builds upon the results of this empirical study. The fundamental results of this research study have only been based on project managers' points of view. Similar research in this area can be conducted (and is strongly recommended) by asking questions to both project managers and team members. Since project managers sometimes may not want to be honest about themselves, essential information regarding project management processes of a particular project can be gathered from team members. This data collection could determine a more holistic view and could probably reveal more pitfalls and misconceptions in innovation projects regarding positive and negative influences of collective creativity and teamwork. Moreover, efforts should be put on finding other anti-patterns in innovation projects due to lack of any other research on this matter. The results from such research could also benefit risk analysis and human resources management along with time management and project management in general.

References

[Addis, 2009] Scott Addis, Creativity and innovation. *Rough Notes,* **152** (4), 2009, 78-81.

[Akyord, 1996] Michael Akyord, Anti-patterns session notes. In: *Proceedings of the Object World West*. San Francisco, 1996.

[Alexander *et al.*, 1977] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl, and King Shlomo Angel, *A Pattern Language: Towns, Buildings, Constructions.* Oxford University Press, 1977.

[Ambler, 2015] Scott W. Ambler, *Ambysoft*. Retrieved January 25, 2015, from The "Change Prevention Process" Anti-pattern: http://www.ambysoft.com/essays/changePrevention.html

[Belding, 2014] Jeff Belding, *Project Management and Innovation*. Retrieved December 15, 2014, from Barnes & Conti: http://www.barnesconti.com/articles/innovativeProjectMgmnt.html

[Raptopoulou, 2012] Charikleia Raptopoulou, Eleni Berki, Timo Poranen, Ioannis Stamelos, and Lefteris Aggelis, Management anti-patterns in finnish software industry. In: *Proceedings of the Software Quality Management*, 173-187.

[Blackwell, 2010] Alan Blackwell, Lee Wilson, Charles Boulton, and John Knell, Creating value across boundaries: Maximising the return from interdisciplinary innovation. NESTA, Research report, May 2010. Also available as http://www.nesta.org.uk/sites/default/files/creating_value_across_boundaries.pdf

[Brooks, 1979] Frederick Brooks, *The Mythical Man-Month.* Addison-Wesley, 1979.

[Brooks, 1995] Jr. Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition.* Addison-Wesley, 1995.

[Brown *et al.*, 1998] William H. Brown, Raphael C. Malveau, Hays W. McCormick, and Thomas J. Mowbray, *Anti-patterns: Refactoring Software, Architecture, and Projects in Crisis.* Wiley Computer Publishing, 1998.

[Brown *et al.*, 2000] William J. Brown, Hays McCormick, and Scott W. Thomas, *Anti-patterns in Project Management.* Wiley Computer Publishing, 2000.

[BSSC, 1995]. BSSC, *Guide to Software Project Management.* ESA Publications Division.

[CMS, 2005] CMS, *Selecting a Development Approach.* Retrieved October 17, 2014, from Centers for Medicate and Medicare Services (CMS): http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf

[Coplien, 1994] James Coplien, Object World briefing on design patterns. In: *Proceedings of the AT&T Bell Labs Conference Tutorial.* San Fransisco.

[Cunningham & Cunningham, 2011] Cunningham & Cunningham, Inc., *Rule Of Three*. Retrieved November 20, 2014, from http://c2.com/cgi/wiki?RuleOfThree

[Demola, 2015] Demola, *Completed and ongoing projects*. Retrieved January 20, 2015, from Demola: http://tampere.demola.net/projects/6104

[El Emam and Koru, 2008] Khaled El Emam and A. Günes Koru, A replicated survey of IT software project failures. *IEEE Software*, **25** (5), 2008, 84-90.

[Fairley, 2009] Richard E. Fairley, *Managing and Leading Software Projects.* Wiley-IEEE Computer Society Pr, 2009.

[Filippov and Mooi, 2009] Sergey Filippov and Herman Mooi, Innovation Project Management: A Research Agenda. In: *Proceedings of the 6th International Conference on Innovation and Management*, 65-78, 2009.

[Gamma *et al.*, 1994] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Resuable Object-Oriented Software.* Addison-Wesley Professional, 1994.

[Gurteen, 1998] David Gurteen, Knowledge, creativity and innovation. *Journal of Knowledge Management,* **2** (1), 1998, 5 - 13.

[Jimenez, 2006] Edwards Jimenez, *Anti-patterns*. Retrieved November 20, 2014, from http://www.anti-patterns.com/EdJs_Paper/Anti-patterns.html#_Toc133670512

[Jones, 2013] Capers Jones, *Evaluating Ten Software Development Methodologies*. Retrieved October 17, 2014, from Namcook Analytics: http://namcookanalytics.com/evaluating-ten-software-development-methodologies/

[Kapsali, 2011] Maria Kapsali, Systems thinking in innovation project management: A match that works. *International Journal of Project Management*, **29**, 2011, 396-407.

[Kärpijoki, 2001] Vesa Kärpijoki, Anti-patterns. In: *Proceedings of the Tik-110.498 Seminar on Design Patters.* Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, 2001.

[Keegan and Turner, 2002] Anne Keegan, J. Rodney Turner, The management of innovation in project-based firms. *Long range planning* , 367-388.

[Komchaliaw and Wongthongtham, 2010] Surasak Komchaliaw and Pompit Wongthongtham, A state of the art review on software project performance management. In: *Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies*, 653-655.

[Laplante and Neill, 2006] A. Phillip Laplante and Colin J. Neill, *Anti-patterns: Identification, Refactoring and Management.* Auerbach Publications, 2006.

[Lechler and Grace, 2007] Thomas Lechler, Emily Grace, Successful Management of Highly Innovative and Urgent Projects: Analyzing project management practices to reveal strategic directions. In: *Proceedings of the PICMET 2007*, 2049-2056.

[Lewis, 2007] James P. Lewis, *Fundamentals of Project Management* (Vol. 3.). American Management Association, 2007.

[Lunenburg, 2011] Fred C. Lunenburg, Leadership versus management: A key distinction - at least in theory. *International Journal of Management, Business, and Administration* , **14** (1), 2011, 1-4.

[Lyytinen and Robey, 1999] Kalle Lyytinen and Daniel Robey, Learning failure in information systems development. *Information Systems Journal* , **9** (2), 1999, 85-101.

[Mellors, 1996] John Mellors, Managing and leading in the next century. *Australian Journal of Public Administration*, **55** (3), 1996, 83-89.

[Meredith and Mantel, 2009] Jack R. Meredith and Samuel J. Mantel, *Project Management: A Managerial Approach* (Vol. 7). John Wiley & Sons, Inc., 2009.

[Pavitt, 2005] Keith Pavitt,. Innovation Processes. In: J. Fagerberg, and D. C. Mowery (eds.), *The Oxford Handbook of Innovation*. Oxford University Press, 2005, 86-114.

[Peters, 2008] Lawrence J. Peters, *Getting results from software development teams.* Microsoft Press, 2008.

[Phillips, 2004] Joseph Phillips, *PMP Project Management Professional Study Guide* (5. ed.). McGraw-Hill, 2004.

[Pinto and Slevin, 1988] Jeffrey K. Pinto and Dennis P. Slevin, Project success: definitions and measurement techniques. *Project Management Journal*, **19** (1), 1988, 67-72.

[PMI, 2013] PMI, *A guide to the project management body of knowledge (PMBOK® guide).* Pennsylvania: Project Management Institute, 2013.

[PMI, 2013b] PMI, *Communication: The Message is Clear.* PMI, 2013.

[Royce, 1998] Walker Royce, *Software Project Management: A Unified Framework.* Addison-Wesley, 1998.

[Settas, 2011] Dimitrios Settas, Software Project Anti-pattern Knowledge Management. *Doctoral Thesis*. Aristotle University of Thessaloniki.

[Shvets *et al.*, 2015] Alexander Shvets, Gerhard Frey, and Marina Pavlova, *SourceMaking*. Retrieved January 25, 2015, from Project Mismanagement: http://sourcemaking.com/anti-patterns/project-mismanagement

[Dennis and Slevin, 1988] Dennis P. Slevin and Jeffrey K. Pinto, Critical success factors across the project life cycle. *Project Management Journal*, **19**, 1988, 67-75.

[Söderlund, 2004] Jonas Söderlund, Building theories of project management: past research, questions for the future. *International Journal of Project Management* **22**, 2004, 183-191.

[Squalli and Wilson, 2014] Jay Squalli and Kenneth Wilson, Intelligence, creativity, and innovation. *Intelligence*, **46**, 2014, 250–257.

[Stamelos, 2010] Ioannis Stamelos, Software project management anti-patterns. *The Journal of Systems and Software*, **83** (1), 2010, 52-59.

[Tomala and Senechal, 2004] Frederic Tomala and Olivier Senechal, Innovation management: a synthesis of academic and industrial points of view. *International Journal of Project Management* **22**, 2004, 281-287.

[Vargas and Garcia-Santillan, 2011] G. José Vargas and Arturo Garcia-Santillan, Management in the Innovation Project. *Journal of Knowledge Management, Economics and Information Technology*, **7**, 2011, 1-24.

[Wallin and Land, 2012] Christina Wallin and Rikard Land, *Software Development Lifecycle Models: The Basic Types*, 2012.

[Webster, 1995] Bruce Webster, *Pitfalls of Object−Oriented Development.* M&T Books, 1995.

[Wikipedia, 2014] Wikipedia., *Project management*. Retrieved November 15, 2014, from Wikipedia: The Free Encyclopedia: http://en.wikipedia.org/wiki/Project_management#Project_managers

[Yusuf, 2009] Shahid Yusuf, From creativity to innovation. *Technology in Society*, **31**(1), 2009, 1-8.

[Zwikael and Globerson, 2004] Ofer Zwikael and Shlomo Globerson, Evaluating the quality of project planning: a model and field results. *International Journal of Production Research* , **42** (8), 2004, 1545-1556.

APPENDIX A - Questionnaire

Research on Project Management Anti-patterns

Dear project manager,

I am Dilem Aydinli who is a Software Development M.Sc student in the University of Tampere (Finland). I am currently writing my thesis named "Project Management Anti-patterns in Software Innovation Projects". I am conducting this survey for the research purpose of my thesis.

Anti-pattern is a form of pattern which describes commonly occurring solutions to a problem that generates negative consequences. The aim of this questionnaire is to find out project management anti-patterns in innovation projects. I would like you to help me reach this objective. While answering questions, you don't need to know what an anti-pattern is. The questions are very simple and the only thing you have to do is giving accurate answers to the questions and be honest.

The questionnaire consists of two main parts as general and research based questions. In the first part of the survey, it is expected to answer questions about the project manager, examined project and team. Second part consists of research based questions. The questionnaire will take maximum 20 minutes of your time.

All information to be collected will remain confidential and be used for academic purposes only. If you have any concerns regarding privacy issues or questions, do not hesitate to contact me at dilemaydinli@gmail.com.

Thank you for your time and collaboration,

Best regards,
Dilem Aydinli

Supervisors

1. Eleni Berki
Assistant Professor at School of Information Sciences, University of Tampere
Eleni.Berki@uta.fi

2. Timo Poranen
Lecturer at School of Information Sciences, University of Tampere
Timo.T.Poranen@uta.fi

* Required

Manager's General Information

1. What is your gender? *

- ○ Male
- ○ Female
- ○ Other
- ○ Don't want to answer

2. How old are you? *

- < 20
- 20-25
- 26-30
- >= 30

3. What is your nationality? *

4. How many innovation projects have you completed as a project manager except the examined project? *

0	1	2	3	4	5	6	7	8	9	10
○	○	○	○	○	○	○	○	○	○	○

General Questions About the Examined Project

5. Name of the project *

6. What was the degree of innovation in your project? *

from 0 - no innovation at all to 5 - completely new product/innovative way of working

0	1	2	3	4	5
○	○	○	○	○	○

7. To what degree was your project successful? Please choose from the list. *

8. Which project development method was adapted for the examined project? *

- No use of development model
- Waterfall Model
- Agile Development
- Iterative and Incremental Development
- Spiral Model
- Other:

9. Were you able to finish the project at the project deadline? *

- ◯ Yes, all promised requirements were implemented.
- ◯ No, 75% of promised requirements were implemented.
- ◯ No, 50% of promised requirements were implemented.
- ◯ No, 25% of promised requirements were implemented.
- ◯ No, there is no working product at all.

General Questions About the Team

10. How many people were in the team excluding project manager? *

1	2	3	4	5	6	7	8	9	10
◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

11. How many of them participated in an innovation project before? *

0	1	2	3	4	5	6	7	8	9	10
◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

12. Were there people from different disciplines in the team? *

- ◯ No
- ◯ Yes

13. If you answered "yes" to the previous question, what were the disciplines?

13. a. Were there any conflicts between people who have different backgrounds (e.g. cultural, national, organizational etc.) What was the reason of that conflict?

13. b. Did you have any difficulties because of interdisciplinarity? How did you manage this situation?

Part 2 - Research Based Questions

1. What kind of schedule did you follow through the development of the project? *

- ◯ It was a carefully planned schedule.
- ◯ The schedule was flexible.
- ◯ Other:

2. During your project, did you spend more time with some task(s) than was planned? *

- ◯ No
- ◯ Yes

3. If your answer is "yes" for the second question, which task did you spend more time than you planned?

You can select more than one answer.

- ☐ Initiation
- ☐ Analysis
- ☐ Planning
- ☐ Design
- ☐ Implementation
- ☐ Testing
- ☐ Closing
- ☐ Other:

4. If your answer is "yes" for the second question, why did you encounter the delay in the project schedule?

5. If your answer is "yes" for the second question, did this situation cause serious problems in the project? How did you manage the situation?

6. Did you produce plans (e.g. specifications, architecture and risk management plans) at sufficient level? *

- ◯ Yes, plans helped a lot.
- ◯ Plans were insufficient.
- ◯ There was no proper plan at all.
- ◯ Other:

7. Did you employ specific control and monitoring? *

- ◯ Yes, code reviews and software inspections were evaluated frequently.
- ◯ Random code reviews and minimal software inspections were done.
- ◯ No, I was not able to control and monitor the project.
- ◯ Other:

8. If there were some problems because of inadequate planning, did inadequate plan, control and monitor of the project cause any problems in the project? If "yes", please explain what kind of problems they were.

e.g. there might be some problems in design and testing phase because of insufficient architectural plans and code reviews.

9. How frequently did the project's requirements change? *

- ◯ Always
- ◯ Often
- ◯ Occasionally
- ◯ Rarely
- ◯ Never
- ◯ Other:

10. How did you manage changes in requirements?

11. What was the main source of change in requirements? *

- ☐ Clients' demands
- ☐ Team members' ideas
- ☐ Technical reasons
- ☐ Schedule slip
- ☐ Team motivation
- ☐ Other:

12. How do you as a project manager engage team creativity into your innovation projects ? *

Definition of team creativity: It is the creation, development, evaluation, and promotion of novel ideas in groups.

13. Did you feel that creativity of team workers were undermined at some point? *

If your answer is "yes", you can select more than one answer.

- ☐ Yes, because of too much customer involvement.
- ☐ Yes, because of the tight schedule.
- ☐ Yes, because of excessive project management.
- ☐ Yes, because of working environment.
- ☐ No.
- ☐ Other:

Free comments on managing innovation projects. *

APPENDIX B – Project managers' comments on innovation projects

- "*The commitment of all team members is essential for a successful project of any kind, especially in this kind of projects. Language skills are also a must since they are necessary to develop any ideas. I honestly believe that my team would have worked and performed better if we had been 3 instead of 5 in the beginning. The inexperience and lack of commitment from the developers only delayed the project. Sometimes less is more.*"
- "*I had the privilege to work with a team that was motivated and interested in the project. However, I also noted that after it was decided that I was the project manager, some of the other team members really expected me to tell them exactly what needs to be finished-so even though they were all motivated, they still needed direct assignments and tasks. Only one of them worked on things that were related to our project but weren't directly assigned by me. One of the team members actually wrote in the final report that as soon as I stopped asking what everybody will be doing before next week (which was my style at the beginning of the project), things started to slow down. So for future reference I would say that even though the team utilizes democracy in decision making, one cannot undermine the role of a project manager.* "
- "*Human psychology is essential for understanding and controlling human behavior.*"
- "*It was a good experience and it taught me a lot from myself and managing a project. After this I feel like this is what I can do if needed, but I am not so sure if I want to do it again. Taking responsibility is the most important thing when being a part of this and also to be able to cope with pressure, motivation shortage, deadlines and uncertainty.*"
- "*It is a good experience for me. Working with people from different background improved my communication skill and we really got many valuable ideas.*"
- "*It is necessary that the team believes in the project and possibilities*"
- "*It is not an easy task, especially if you haven't studied exactly product development or process development. Luckily, one of our members was in this field and he helped as a lot.*"
- "*In my experience the most difficult part in the innovation projects are to get people motivated to work on a common goal. There are always some team members who are very motivated and work hard on a project, and then there are those who don't have the time needed or don't care that much. Adding cultural differences to that will result in somewhat interesting conflicts and those teams who can overcome that will get the project done and those who can't will most likely fail.*"