

Timo Poranen (ed.)

**Software Project Management
Summaries 2009**



DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TAMPERE

D-2009-8

TAMPERE 2009

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER SCIENCES
SERIES OF PUBLICATIONS D – NET PUBLICATIONS
D-2009-8, DECEMBER 2009

Timo Poranen (ed.)

**Software Project Management
Summaries 2009**

DEPARTMENT OF COMPUTER SCIENCES
FIN-33014 UNIVERSITY OF TAMPERE

ISBN 978-951-44-7951-9
ISSN 1795-4274

Preface

This report contains summaries of project management articles published in international scientific journals and conferences. The summaries were written as a compulsory task for the Theory of Software Project Management –course held fall 2009.

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

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

Timo Poranen

Tampere, December 2009

Preface.....	1
Software project management using PROMPT: A hybrid metrics, modelling and utility framework.....	4
Business-oriented software process improvement based on CMMI using QFD	6
Do project managers' leadership competencies contribute to project success?	8
Icarus' predicament: Managing the pathologies of overspecification and overdesign.....	10
Stakeholder analysis in projects: Challenges in using current guidelines in the real world.....	11
E-ethical leadership for virtual teams.....	13
The effect of decision style on the use of a project management tool: An empirical laboratory study.....	15
Empirical studies of agile software development: A systematic review	17
Success in IT projects: A matter of definition?.....	21
A survey study of critical success factors in agile software projects	23
Critical skills for IT project management and how they are learned.....	25
Effect of organisational position and network centrality on project coordination	27
Using planning poker for combining expert estimates in software projects (1/2)	29
Using planning poker for combining expert estimates in software projects (2/2)	31
Experience on knowledge-based software engineering: A logic-based requirements language and its industrial applications	32
An Introduction to agile methods.....	34
A practical use of key success factors to improve effectiveness of project management	36
Reducing software product development time.....	37
A project contingency framework based on uncertainty and its consequences	39
Does risk management contribute to IT project success? A meta-analysis of empirical evidence.....	40
Use and benefits of tools for project risk management.....	42
Improving project outcomes through operational reliability: A conceptual model	44
Global virtual teams for value creation and project success: A case study	46
Managing user expectations on software projects: Lessons from the trenches	

.....	47
The interactive effect of team dynamics and organizational support on ICT project success	49
Top management support: Mantra or necessity?	50
Why projects fail – How contingency theory can provide new insights? A comparative analysis of NASA’s Mars Climate Orbiter loss.....	52
Software project management anti-patterns	54
Why and how can human-related measures support software development processes?	56
The impact of software process standardization on software flexibility and project management performance: Control theory perspective	57
Foundations of program management: A bibliometric view	59
A feasibility evaluation on the outsourcing of quality testing and inspection ...	61
Towards a conceptualisation of PMO’s as agents and subjects of change and renewal	62
How do personality, team processes and task characteristics relate to job satisfaction and software quality?	63

Software project management using PROMPT: A hybrid metrics, modelling and utility framework

David M. Raffo, Information and Software Technology, volume 47, pages 1009-1017, 2005

Background

Decision making in software processes can be difficult, even if the process is clear and the programmers are aware of the practices. The paper presents a forward-looking decision support framework to help managers predict likely outcomes and bring stray projects back on track. The method is called PROMPT (PROject Management of Process Trade-offs). PROMPT measures trade-offs among performance using outcome based control limits (OCBLs).

Results

The PROMPT method is an iterative process that augments PLAN-DO-STUDY-ACT cycle by utilizing in-process data with quantitative models. Models support the planning phase by estimating the financial outcome of each decision alternative. The progress of the plan is studied by applying timely metrics to model parameters. Models in studying phase predict distributions of project performance measures. The trade-off is whether to continue as planned or to take corrective actions. When corrective action is necessary, the model is used to identify the best corrective action from all possible choices.

PROMPT can be used when model:

1. Enables managers to use OCBLs to set targets and acceptable ranges to one or more performance measures that are of interest to managers.
2. Uses OCBLs to predict performance as point values or as stochastic distributions with a mean and standard deviation.
3. Uses Process Simulation to capture sufficient level of process detail to identify process level issues.
4. Is capable of updating it's parameters by utilizing up-to-date data from on-going project.

The paper presents a real world example in which three performance measures were set: cost, schedule and quality. Performance measures were estimated using past project data and the estimates were used in the bid process. A discrete event simulation model was developed to predict overall project performance by using actual project data when available and by interviewing managers and senior programmers. Several inputs were assigned to the model to capture data timely.

First management sets the OCBLs, which are acceptable ranges or control limits to performance measures. Before starting the project, but after the preliminary design phase, predictions are compared against OCBLs to estimate probability to stay within limits.

Link between model parameters and up-to-date project metrics is critical to improve accuracy of estimated project outcomes. Repository that facilitates a real-time project data is essential. In the example project a repository based on 'transformation view' was used. Transformation model transforms artifacts to other type of artifact, like a

design artifact to a code artifact. The repository is automatically updated and a snapshot can be taken at the completion of each iteration to update model parameters.

Updated parameters are used to refine predicted outcomes, which are then compared against OCBLs. Corrective action needs to be taken whenever OCBLs are not met.

When corrective action is decided to be taken, there are a variety of potential actions that can be evaluated using the simulation model. Results of actions are compared to the OCBLs to determine the best corrective action. To determine the best action, an utility function with some additional criteria (such as staffing and schedule constraints) must be developed. In the example an utility function was created by converting implementation cost, remaining defects and effort to their cash equivalents. Function was used to estimate a trade-off between cost and schedule to provide highest benefits while keeping costs to a minimum. After the corrective action is taken the model parameters are updated again with actual observations from the design.

Conclusions

In the software developing process many complex and challenging decisions have to made. The paper presents a real-life application of the PROject Management Process Trade-off method. Collecting metrics to a flexible repository is explained by an example. Metrics are applied to a simulation model to predict outcomes. Predictions are compared to the Outcome Based Control Limits to determine need of corrective actions. Utility function are developed to evaluate best corrective action. The method supports the decision making in software development process and the ability to quantitatively monitor the software projects.

Antti Torkko

Business-oriented software process improvement based on CMMI using QFD

Yan Sun and Xiaoqing Liu, Information and Software Technology, volume 52, pages 79-91, 2010

Background

The Capability Maturity Model Integrated (CMMI) is fusion of several models that can be used to assess the capability and maturity level of the processes used by an organization, and to guide the software process improvement (SPI) activities to reach the next level of the maturity. The capability maturity is estimated using a 5-level scale, where level 1 is the beginner level (with no processes defined). The CMMI defines 22 different key process areas (PAs), and provides examples of the best Practices for each PA. A certain organization may have reached maturity level 4 in one of the PAs, but may be down at level 2 in some other area. Organization can be assessed and appraised for their process maturity level, and being CMMI compliant is often regarded a sign of the organization focusing on process issues and thus, providing better product quality.

Besides just being “CMMI compliant”, an organization also needs to ensure its SPI actions are linked to the actual business needs and can provide some tangible benefits to the organization. The article presents a practically explained SPI framework to combine business perspective to the CMMI model by using Quality Function Deployment (QFD) methodology to bridge the gap between business requirements and process improvement. The QFD provides a set of tools and methods - such as the House of Quality matrix - that are designed to prioritize spoken and unspoken customer needs, and to transform those needs into actions, technical designs and specifications. The presented SPI framework is especially targeted for large, complex organization in which diverse stakeholders may have different viewpoints about the most crucial SPI actions.

Results

In the presented SPI framework, the business requirements are first prioritized by criticality. As the QFD is then used to map requirements to the CMMI PAs and Practices, the set priority order will also be reflected in the resulted SPI actions. If resources are scarce, an organization can then choose to implement only those actions that provide most value, i.e. those that best correlate with the business requirements. Alternatively, organization can choose to assign more resources to complete the most important actions.

In Phase 1 (Requirements Elicitation and Integration) of the framework, the requirements from all the different stakeholders groups – such as business executives, R&D managers and quality specialists – are gathered, and the correlation between each of the requirements is identified, using the Requirements Impact Matrix (RI Matrix). The more related a requirement is to other requirements, the higher priority will be assigned. These final priority values of the requirements are then comparable across different perspectives, and the requirements can be integrated into one single set and used in the rest of the SPI framework.

In Phase 2 (CMMI PA Prioritization, or, CMMI Goal Prioritization), the requirements are mapped to the relevant PAs, using the Requirements-Process Areas Impact Matrix (RPA Matrix. For example, requirement "Reduce Cost of Development" could be linked with the "Supplier Agreement Management" area. The priorities of the PAs are then calculated based on their correlations with the requirements, and the priority values of the requirements.

In Phase 3 (CMMI Practice Prioritization), the requirements are mapped to the Practices using the Requirements-Practices Impact Matrix (RPr Matrix). Note that only the Practices presented at the desired maturity level of the identified PA, are considered. For instance, requirement "High conformance to standards" could be linked with Practice "Manage Configurations". The priorities of the Practices are then calculated based on their correlations with the requirements, and the priority values of the PA they belong to, which are now also reflecting requirements priorities.

In Phase 4 (Action Plan Development and Prioritization), the Action Plan House of Quality Matrix (AP-HoQ) is used to derive a set of SPI actions from the prioritized Practices, at the desired capability level of the PAs. The priority values of the SPI actions can be calculated based on the priority values of the Practices, as well as their correlations with these Practices. The actions now state, what needs to be executed in order to reach a particular capability level of a particular PA, and can be used to guide the process improvement in the organization.

The framework can be used with both continuous and staged CMMI. When the continuous CMMI is used, the targeted capability level for a particulate PA's, considered in Phase 2, can be selected freely; i.e. organization can choose to follow best Practices from several different maturity levels. In staged CMMI, the organization is committed to strive to reach a certain capability level in all of the PAs.

Conclusions

SPIs should be planned to satisfy the needs of at least some stakeholders; in the presented SPI framework, the merged stakeholder requirements are the basis for the SPI actions prioritization. The business related process requirements are linked to the CMMI PAs and Practices, and through these links, software development organizations can see the direct benefit to the business in trying to reach a higher level in the CMMI.

Reeta Karjalainen

Do project managers' leadership competencies contribute to project success?

Linda Geoghegan and Victor Dulewicz, Project Management Journal, volume 39, number 4, pages 58–67, 2008

Background

The study explores whether leadership competencies have a statistically significant effect on project success, and if so, which specific leadership skills would contribute to it. Leadership may be defined as being a combination of skills and knowledge. Some skills are personal, such as personal characteristics and emotional intelligence, but some skills are easier to be learned, such as managerial competencies. The study compares the competencies of project managers to the success of their projects.

Results

The study was conducted in a large financial services company in the United Kingdom. The data for this quantitative study was gathered from two sample groups, namely project managers and project sponsors, which were related through common projects. Of the initial sample of 80 project managers a subgroup was selected suitable for the study. The requirements were that the manager has managed a project with a budget larger than £350,000 and has at least four years of experience in project management.

Leadership skills were categorised in to three dimensions: intellectual competence (Critical analysis and judgment, Vision and imagination, Strategic perspective), emotional intelligence (Self-awareness, Emotional resilience, Intuitiveness, Sensitivity, Influencing, Motivation, Conscientiousness) and managerial competence (Managing resources, Engaging communication, Empowering, Developing, Achieving). These skills form the basis of an LDQ (Leadership Dimension Questionnaire) to which project managers answered. LDQ scores were then compared to PSQ (Project Success Questionnaire) scores of the company surveyed.

Factor analysis was used for PSQ's 12 success factors. Three factors were identified: usability, value of project outcome and project delivery. The second factor was not considered relevant for this study and was not discussed further.

Out of the leadership competencies vision, strategic perspective and achieving were least related to project success. Visioning had the lowest score, yet it should be noted that possibly the projects are visioned and strategies planned already in a higher level of the organisation, thus less rope is given to the project manager in these aspects. Manager's competency for critical analysis was the only one that appeared to significantly correlate to project success in the intellectual competence group.

In the usability factor three managerial competencies (managing resources, empowering, developing) and one emotional competence (motivation) were considered highly significant. Four others were considered significant: critical analysis (intellectual competence), influencing, self-awareness and sensitivity (emotional competencies). In the project delivery factor only two managerial competencies proved significant, they were managing resources and empowering.

Conclusions

Based on the findings, there exists a link between managerial competencies and project success. Five managerial, four emotional and one intellectual competence were identified to be significant for project success. Both managerial and emotional skills may explain variations between outcomes of different projects.

While the strengths of the project managers have been identified in the LDQ analysis and the significant competencies have been found by combining LDQ and PSQ data, the information could well be used to improve project success by identifying the weaknesses in project managers' competencies and then putting effort to enhance the shortcomings considered important to the business.

Joonas Mäkinen

Icarus' predicament: Managing the pathologies of overspecification and overdesign

Alex Coman and Boaz Ronen, International Journal of Project Management, 2009, to appear

Background

Overspecification is a phenomenon where too excessive set of requirements is defined for the product. This can happen for example when the marketing does not fully understand the customer needs. Overdesign occurs when developers add unspecified features, performance and technological improvements into product.

Results

Coman and Ronen presented ten case studies from different organizations, where overspecification and overdesign occurred. Implications from case studies indicated that, too ambitious design plans resulted in time delays, bad usability, out-of-focus products and other crisis situations, even bankruptcy of the organization.

The over specification and overdesign results from organization inability to prioritize what are projects and features that bring most value to users and the company. The lack of proper roadmaps and vision about the future can result in making features just for sure. In addition, the organization culture and the development personnel ambitions can result in wasting resources into work that is out of scope. Goldratt's conflict-resolution-diagram were used to analyze the conflicts that lead to the over specification and over design from the marketing and R&D point of view.

The countermeasures start from the customer understanding and leadership ability to make proper decisions. Understanding the core features that bring most value to the users and gradually improving & evaluating the product help focusing resources correctly. The product creation process should be controlled so, that at certain point features are frozen and engineering is properly managed in order to avoid unspecified features. The R&D personnel eagerness to overdesign can be eased by proving possibility to participate more challenging projects from time to time.

For the business leadership level self assessment were provided, which can be used to reveal if the overspecification and overdesign exists within organization.

Conclusions

Overcoming the phenomena of overspecification and overdesign is a critical success factor for organizations. The paper defined the phenomena, root causes provided a set of countermeasures were presented. Authors suggest that further studies should be made in order to measure the wasted resources and model the phenomena further.

Vesa Huotari

Stakeholder analysis in projects: Challenges in using current guidelines in the real world

Anna Lund Jepsen and Pernille Eskerod, International Journal of Project Management, volume 27, issue 4, pages 335-343, 2009

Background

Stakeholder management is considered important in both general management domain and project management domain. However, the study of stakeholder analysis is still in a theoretic stage. There are still challenges in utilizing the current theoretic guideline in real world projects. In this paper, the authors talk about the experiences with four different project groups, and also the problems they encountered within the whole procedure.

Results

To begin with, the authors present the current analyzing guidelines for stakeholder management, which encompasses three important parts, including identification of the important stakeholders, characterization of the stakeholders, and strategies to use to influence each stakeholder. In the activity of identifying stakeholders, conducting brainstorm would be a proper method. And attributes of each stakeholder, such as interests, expectations, power, etc, will be listed. In activity of characterizing stakeholders, project managers should make explicit the needed contribution from stakeholders, the expectation of stakeholders and the power stakeholders have. Within this stage, a stakeholder-commitment matrix will be utilized to record the attitudes of each stakeholder and the expecting attitudes from the managers. And the activity of selecting strategies must rely on the first two activities.

Subsequently, the whole research process is presented. During the 4-month research, authors followed four separated projects and worked with project managers as process facilitators in order to lead project managers to carry out stakeholder analysis according to the guidelines. To guarantee the accomplishment of stakeholder analysis, authors held, with project managers, four meetings, during which authors tried to conduct project managers onto the right track of obtaining information from stakeholders and to feedback existing problems and experiences.

Based on the research, the authors discover several challenges and problems that the project managers encounter in different stages of stakeholders analysis. The challenges and problems are listed below.

- a) Identifying the stakeholders.
 - 1) Project managers are not sure whether they category stakeholders in a right way.
 - 2) Project managers are confused whether to trade some stakeholders as a group or not.
 - 3) Project managers consider it impossible to make an exhaustive list as stakeholders may change somehow.
- b) Determining contributions

- 1) It's hard for project managers to list all the needed contributions.
 - 2) Tacit contribution is hard to assess in advance.
 - 3) It's hard to decide whether to single out stakeholders to other groups.
- c) Determining expectations and benefits
- 1) Project managers do not know what they would acquire from stakeholders according to current guidelines.
 - 2) It's hard to carry out interview strategy to gather information.
 - 3) Project managers are reluctant to contact stakeholders against projects.
- d) Strategy to use
- 1) It's hard for project managers to consider all stakeholders
 - 2) Project managers are forced to make decision without adequate information.

Among these problems and challenges, authors point out that capability to gather information, determination of stakeholders' importance, stakeholders sharing, investigation of stakeholders' attitude and capability to foresee are the main problems that authors discover in the research.

Conclusions

To sum up, plenty of challenges exist when project managers carrying out current guideline in real projects. Research should be done on how to identify important stakeholders, and how to retrieve information from all types of stakeholder and influence them into positive sides of projects. Moreover, the guidelines should be more practical to conduct managers in real projects. And the stakeholder analysis process should be more like an iterative process than a task which has to be done. However, the importance of stakeholder analysis to the projects is beyond doubt.

Xiaozhou Li

E-ethical leadership for virtual teams

Margaret R. Lee, International Journal of Project Management, volume 27, pages 456-463, 2009

Background

The difference between the traditional project team leadership and the leadership for virtual project teams is very important, by studying different theories managers of virtual teams will be able to provide valuable and positive ethical management that will help avoid risks, problems concerning privacy, lack of trust and other unethical behaviors.

Results

The paper provides a literature review on ethical theories related to ethical leadership for virtual project management as well as the challenges of having a virtual team and some solutions.

Results show that project managers should be managers and leaders. Therefore project managers should emphasize ethical behavior and eliminate conditions that might be unethical; these unethical behaviors at the end will carry additional costs to the project and should be avoided.

The first ethical theory related to virtual project management is the Theory Y is part of the participative management approach, in this theory leaders will delegate decisions and job responsibilities as needed, employees will then accept the responsibility of their work and will support the organization by searching for methods to achieve these goals. What this says is that by giving the chances to everyone to participate on the decision making they will be more likely to accept the decisions and consequences.

Another theory presented is the Kantian ethics that states that individuals are treated with respect and never exploited because doing this will be morally wrong. In this theory the most important things to remember is motivation and trust. Results in the article shows that team members expect that their efforts will be reciprocated and not exploited by other members of the team, so if people inside a team have trust they will be more motivated to help each other.

The communitarian ethics focus on community and shared values not from the individual but in the relationship with the community. In this theory we can have two approaches, the first one called the ethic of care in which we should take into consideration people near us and relationships, as tolerance in virtual communities or moderators in online forums. The other approach is called egalitarianism, where all individuals are equal even in personal areas such as politics, religion, economics, this could be negative in virtual environments because people want to show and express their ideas and this approach will limit this freedom.

There are very important issues concerning ethics in virtual teams like unethical use of sensitive material and unethical behaviors these can be avoid if project manager promotes a caring and trusting environment within all members of the team.

Another to consider are superficial codes of conduct, this can be avoid if all members of the team participate on it, this will bring a sense of community and trust needed

specially in virtual teams where no face-to-face meetings take place and miles of distance can create aggressive behavior.

An important issue found on virtual community not found in non-virtual teams is the social isolation, this may appear as a need for a caring environment, motivation and loyalty from the team resulting into lack of commitment and communication from one or more members of the team. Project manager should take special care on individuals and their identity as a team member in the virtual world.

Ethical behavior is one of the first things that compromised a high-risk environment, the use of software programs with a policy-based review are sometimes consider only as effective as information about ethics is entered into it and can be seen as a lack of trust between manager and members, strong leadership must support this in order to succeed.

In order to create a caring environment project manager should discuss about personal goals and expectations of each member of the team, communication is a key factor that can help to reduce many ethical problems.

Lack of stakeholders involvement should be carefully consider, the challenge of all teams is to influence the requirements to ensure a successful project and reduce stakeholder dissatisfaction.

Conclusions

Trust is the major contributor of motivation and positive relations in a virtual team. Project managers should regulate ethics in virtual teams, motivate and give individuals the chance to participate in the decision making process, they should bring to the team the sense of community that is strongly needed while working on geographically distant locations. And finally avoid spying employees as it violates trust and privacy.

Project managers always have to deal with the rapid changes and uncertainty of a project, but by understanding the different needs of management in virtual teams they will be able to create a more ethical environment were all parties involved will benefit from it and have interest on its success.

Georgina Monjaraz Gomez

The effect of decision style on the use of a project management tool: An empirical laboratory study

Terry L. Fox and J. Wayne Spence, The DATA BASE for Advances in Information Systems, volume 36, number 2, pages 28-42, 2005

Background

Numerous project management tools exist and their use is considered highly beneficial to the success of a software development project. The tools can, for example, help to determine whether a proposed schedule is realistic, and whether the necessary resources for the successful completion of the project are available. However, despite availability of multitudes of tools, they all seem to be designed based on the assumption of a systematic, structured, analytic approach to project management. Project managers, on the other hand, have been found to be diverse in their preferred approach to decision making and problem solving. There is a concern that project managers cannot function at their best capacity when forced to use a tool that does not support their preferred style of decision making.

Results

An empirical laboratory study was conducted to study the effect of the preferred decision making approach of a project manager on his or her performance on a set of computer-assisted project management tasks. A total of 52 participants were recruited from eight large US organizations. The preferred decision making approach of a participant was determined by administering him or her the academically well regarded Decision Style Inventory (DSI) questionnaire. Each participant was asked to complete the same set of four project management tasks related to a provided sample scenario. The tasks were completed using the industry-standard project management tool, Microsoft Project. All of the participants had previous experience in working with the tool. The task completion times and the number of deviations from the ideal solutions were recorded, and a statistical analysis was performed.

Three statistically significant correlations relevant to the research question were found. The participants classified as having a dominant Directive, Analytical or Conceptual decision style completed the tasks faster than the participants with a dominant Behavioral decision style. The difference was statistically very significant. The participants with a dominant Directive or Analytical decision style had developed more complete initial project plans than the participants with a dominant Conceptual or Behavioral decision style, and they were better able to identify the variances in a project plan. These differences were statistically somewhat significant.

Conclusions

The results suggest that a project management tool assuming a very structured decision making approach is less able to support project managers preferring a more unstructured approach. A tool better affording hands-on, direct manipulation of content might improve the performance and personal satisfaction of these users. However, studies with more participants are needed to validate the findings of this study, and even then, the recognition of the need to support multiple cognitive

approaches is just the first step in the process of providing tools to support them.

Aapo Laitinen

Empirical studies of agile software development: A systematic review

Tore Dybå and Torgeir Dingsøy, Information and Software Technology, volume 50, issues 9-10, pages 833–859, 2008

Background

This is a systematic review of agile software development research, before which the same kind has not previously been published. It seeks to evaluate, synthesize, and present the empirical findings on agile software development up to and including 2005, and provides an overview of topics researched, their findings, strength of the findings, and implications for research and practice.

Results

At the very beginning, the researchers developed a protocol for the systematic review by following the guidelines, procedures, and policies from different organizations and consultation with software engineering specialist on the topic and methods. This protocol specified the research questions, search strategy, inclusion, exclusion and quality criteria, data extraction, and methods of synthesis. Thus, the search strategy identified 1996 studies, of which 36 were identified as empirical studies.

Then these empirical studies were categorized into four themes: (1) introduction and adoption, (2) human and social factors, (3) perceptions on agile methods, and (4) comparative studies. To assess each study, researchers develop a summery of the questions according to 11 quality criteria based on the Critical Appraisal Skills Program (CASP) and by principles of good practice for conducting empirical research in software engineering. And also grading of the each 11 criteria was done on a dichotomous (“yes” or “no”) scale.

The review shows that many promising studies of the use of agile methods have been reported. Although serious limitations have been identified, e.g., that the role of on-site customer seems to be unsustainable for long periods and that it is difficult to introduce agile methods into large and complex projects, the results of the review suggest that it is possible to achieve improved job satisfaction, productivity, and increased customer satisfaction. The strongest and probably most relevant suggestion is a high level of individual autonomy must be balanced with a high level of team autonomy and corporate responsibility. It seems important to staff of agile teams that they have faith in their own abilities combined with good interpersonal skills and trust. Evidence suggests that instead of abandoning traditional project management principles, one should rather take advantage of these principles and combine them with agile project management. The evidence also suggests that agile methods not necessarily are the best choice for large projects. Thus, in ending the study suggest that practitioners carefully study their projects’ characteristics and compare them with the relevant agile methods’ required characteristics.

Conclusions

A clear finding of the review is that we need to increase both the number and the quality of studies on agile software development. However, it is important to recognize that researcher excluded “lessons learned” papers that were based merely on expert opinion. If the review had included this literature, the current study could, in principle, have provided more data. Furthermore, biasness in the selection of publications and inaccuracy in data extraction would create the risk of omitting relevant studies and limit the review.

Avishek Barua

Different alliance relationships for project design and execution

Brian Hobbs and Bjørn Andersen, International Journal of Project Management, volume 19, issue 8, pages 465-469, 2001

Background

There are many ways of managing a project. In the years 1995-1999 IMEC research programme performed test on 60 projects to find out what practices are the most effective. The article is about different approaches to relationship between owner and supplier during design and execution phases. There are four different shapes of such relationships. Two of them are based on arms' length system. Other two shows relational approach. There have been plenty of publications about arms' length relationship, so this paper treat mostly about relational system.

Results

The project can be divided into phases- front end, design and execution. If we use traditional approach in some of the phases, it don't mean we can't use other approaches in different phases.

If front end phase is done by one company and execution part is done by others it mean 'traditional sponsorship' system is used. When front end is done by partner companies and execution is done by some other firms than the system is called 'partners in ownership'. In these approaches suppliers are bidding to gain contract. Before they get it, they have no influence on a project. Work that suppliers have to do is pre-specified which means that owner have very little influence on project after contract is signed. So what is to be provided depend on owners and how it will be done depend on suppliers. This lack of influences between owner and supplier in their phases of work can be seen as drawback.

Solution to it can be seen in relational way of managing project. Owner can take part in execution process. Than we can call it 'partners in design and execution'. If suppliers participate in front end phase we call it 'relational development and execution'.

In 'partners in design and execution' there can be distinguished following types of relationship:

-Partnering: The communication between players is improved by owners representatives which help to solve problems. This results in improved quality, schedule and reduced cost.

-Frame agreements: Owner resign from competitive bidding in favor to familiarity to personnel, resources, management style, skills and knowledge of supplier.

-One-off Integrated project teams: Sometimes owners- large companies- couldn't find feasible way to complete a project. They had to abandon a project or find a different way to construct it. A way to do it was to find specialized suppliers with knowledge to find feasible design. These projects resulted in success.

-Sticky informal networks: Distinguish itself because of informal relation between owner and suppliers. This is based on long cooperation and trust, not on strict contracts.

'Relational development and execution' is characterized by cooperation between some firms through all stages of project. There can be listed following types of such cooperation:

-Project subjected to intense scrutiny: Project is intensively analyzed by government agencies, pressure groups, executing firms and private investors other than execution companies to check worst scenarios and find solution best to public interest.

-All integrated business perspective: concessionaire is a group of firms that have to take care of design and cost reduction and project quality. They are responsible for both designing and construction of project.

-Faster and more efficient execution: Presence of execution companies in early stages of project results in improved performance- firms are familiar with plans and they contributed ideas and solutions to make construction process cheaper and shorter.

'Relational development and execution' kind of managing project can be very lucrative to both owner and executor. On the other hand it is risky, because big profits to private company which provided public service can be marked by taxes or even nationalized.

Conclusions

As the conclusion we can see that there is no best way to manage every project. For each of them different approach can bring success, so developers have to decide which practice may be the best to project context.

From owner perspective the safest way to undertake a project is to do it the same as it has been done for years with good effect. The usual way, new practices are found and used is when owners are pressured by environment or perspective of great profit is seen.

Bartłomiej Kutera

Success in IT projects: A matter of definition?

Graeme Thomas and Walter Fernández, International Journal of Project Management, volume 26, pages 733-742, 2008

Background

To catch the essence behind project success has been the goal of many experienced project managers, project organizations and project researchers over the years. Although it is widely agreed that project success is a difficult, multi-dimensional, elusive and usually heavily subjective concept. What is not being agreed on is which of these many dimensions best represent success. The goal of the study summarized here was to gain a better understanding of how companies in Australia define and measure IT project success.

Results

This exploratory study followed a qualitative theory-building paradigm and departs from the two following questions:

1. How do organizations in Australia define IT project success?
2. Which methods are most effective and why?

To acquire data, mini-case studies with semi-structured interviews were conducted with 72 senior managers in 36 companies operating in three industries: Finance and Insurance; Mining; and Electricity, Gas and Water Supply. Each company, two interviews were conducted. The interviews investigated evaluation at various stages of the project life cycle, both ex-ante and ex-post. Primary interview, focused on evaluation practices in the company in general, including the company view of success and failure, was with a chief information officer, program office manager or equivalent. The secondary interview was conducted with a project manager and explored the evaluation of a recently completed IT project, whether the project was a success and how success was determined. The researchers also collected and analyzed 362 documents relating to project management and evaluation practices for contextual, informational triangulation purposes.

The companies were first individually analyzed and definitions of success were coded and allocated to categories based on the dimensions of success found in literature. The formality of the success construct and the effectiveness of measurement were rated based on the descriptions provided by the participants. The companies were sorted into categories based on the primary interview participant's rating of confidence that IT projects are producing business benefits. The practices used to define and measure success were then compared across groupings, in particular how the level of confidence related to the formality of the success construct and the effectiveness of the measurement.

Those companies that had a formal success construct, effective measurement of success, and medium to high levels of confidence shared some important common practices. These effective practices were: an agreed definition of success, consistent measurement and use of results.

Companies who were highly effective at measuring success, measured project management success using a post-implementation review and tracked benefits for 6–

12 months after implementation. Continuous measurement of success also allowed companies the opportunity to stop projects in a controlled manner. Companies who used the measurement results effectively were willing to redirect project resources based on the understanding of the relative importance of project success criteria and were willing to stop projects. This resulted in improved project management and better use of resources.

Most companies who had post-implementation review processes made very use of the findings from these reviews to improve processes or learn corporately. Companies where lessons learned were captured, used to drive process improvement and then shared, reported improvements in the estimation and management of their projects.

Conclusions

The key finding of the study was that companies who clearly define and effectively measure IT project success have a greater chance of achieving success. Companies that formally defined success, consistently measured success and acted on the results, had improved IT project outcomes and better utilized project resources.

Janne Pihlajaniemi

A survey study of critical success factors in agile software projects

Tsun Chow and Dac-Buu Cao, Journal of Systems and Software, volume 81, pages 961-971, 2008

Background

Agile software development methods are a relatively new trend in software project management. They are not as thoroughly analysed as traditional methods. The main aim of Chow's and Cao's "*A survey study of critical success factors in agile software projects*" was to reveal critical factors which are the most important for a successful agile project.

Results

Chow and Cao prepared a list of factors affecting the success or the failure of an agile software project. Factors belonged to 5 categories: Organizational, People, Process, Technical and Project. The source of the data was agile literature. Authors used reliability analysis and calculated internal consistency of the set of factors. The analysis reduced the number of independent factors to 12. They were all critical success factors in terms of four success dimensions categorised by authors as the most important: Quality, Scope, Time and Cost.

The next step was distributing questionnaires to 83 Agile Alliance user groups. The questionnaire consisted of questions regarding demographic data, assigning priorities to success factors, prioritising success dimensions and additional comments. 408 responses from individual people from 109 agile projects had been received.

Collected data was analysed using multiple regression analysis. Authors aimed at finding a relationship between success factors and an agile project success.

Results of the regression analysis showed that not all 12 factors were truly critical for successful agile projects. The analysis allowed authors to create a following list of critical factors, ordered by decreasing importance: Delivery Strategy, Agile Software Engineering Techniques, Team Capability, Project Management Process, Team Environment and Customer Involvement.

Another finding was the fact that out of five different factor categories Technical included 5 critical factors, People — 3 critical factors, while Process and Organizational both had one critical factor. None of critical factors belonged to the Project category.

Conclusions

Results of the research agree with the content of the Agile Manifesto. Three most important critical factors revealed by the study resemble issues covered by the Manifesto.

The study does not support though some of assumptions regarding agile projects. According to the results strong executive support, agile-style work facilities, e.g. pair programming stations, and presence of agile-friendly organizational environment, i.e. cooperative culture or universal acceptance of agile methods, are not critical.

Findings of this study may be of significant value for project managers considering usage of agile methodologies. Analysing its results may help to focus on the most important factors of a successful agile software project.

Jan Stepień

Critical skills for IT project management and how they are learned

Hazel Taylor and Jill Palzkill Woelfer, in Proceedings of the special interest group on management information, pages 103-112, 2009

Background

The aim of the research has been to find the answer of the below question:

" What do IT project managers consider to be critical skills necessary for project success, and how they develop those skills."

Methodology applied: Interviewed thirteen experienced IT project managers from organizations by focusing first on what the managers perceived as their most critical project management skills and then on how they had developed those skills.

Objective: To uncover the importance of informal learning channels.

Introduction: To improve the level of management companies rely on manager's performance. And to improve that they depends on management development initiatives. In this research the main focus is on key knowledge and skills that experienced IT project managers view as critical important and how they acquire those critical skills.

Management skills: Nowadays management skills are not tacit, they are available as literature reviews. One of the example is project management body of knowledge (PMBOK). It is the detailed framework of nine knowledge areas, broken down onto activities across five stages or process groups of the project life cycle, that are claimed to encompass the sum of knowledge generally recognized as good practice in the project management profession.

The first step to IT project management is to understand the specific knowledge and skills required. The one way discovered to identifying skills is to first evaluate the work on in-house project. Project management training is the other way thru but it is much time consuming process and not the predictable one. So the best practice discovered is to "let them learn by their experiences".

Research Factors: The main emphasis was in case study investigation by exploration of individual manager's learning. So the main stress factors of this process were:

- Sampling of managers.
- Data collection procedures.
- Analysis procedures.

Results

Critical competencies and learning methods: In the first portion of interview. There are 21 competencies and 85 learning (presented by PMBOK) reviewed on the 13 project manager's responses. On the basis of the number of managers in each competences learning the critical importance factor is determined.

Organizational development interventions: The second portion of interview the focus

is on the learning opportunities the managers had experienced, and how useful they found these learnings. The reviews were held and considered helpful by most of the managers by scaling it on 1-3 scale.

The two types of training experienced highest ratings were:

- Formal project management training.
- Formal leadership skills.

corresponding two most highly mentioned behavioral competencies:

- Concern for order.
- Accuracy and quality, and team leadership.

Conclusions

By having discussions with the managers and by their thoughts. Even after the formal training sessions and the results of the interviews some still believe their own learning techniques and experiments. It is concluded that formal training alone is not enough to foster the development of a particular skill, it may provide necessary foundation for subsequent experiential learning to occur. Encouragement for self reflective practices could be a productive development route. The effective use of a learning tool could occur if they are structured to support and promote the reflection both during and end of the project.

Jaspreet Singh

Effect of organisational position and network centrality on project coordination

Liaquat Hossain, International Journal of Project Management, volume 27, issue 7, pages 680-689, 2009

Background

The 2001-2004 investigation of Enron Corporation led to the court-ordered release of the company's entire email archive. After the trial, efforts by MIT brought the emails into the public domain via an indexed, searchable SQL database. Since then, the archive has become a popular resource for researchers, being the largest real-life dataset from a corporate environment. In this article, the author uses the data to visualize social networks within organizations, revealing relevant relationships within project coordination.

Results

Normally, relationships in a corporate environment are analyzed from a top-down, hierarchical approach. As an alternative to this trend, Hossain introduces a social networking analysis model which emphasizes centrality in organizational communication. He uses a method known as support vector machines to algorithmically filter and categorize the emails, narrowing the scope of the 600,000+ emails to a subset of emails from 80 employees related to a single project (the "Dabhol Power Company"). Each employee becomes an "actor" in the organization, displaying a "willingness" to communicate through the social network. The number of emails sent to other actors strengthens the weights between nodes, revealing a coordination "score" of the individual. This model is visualized as a node-link network, allowing highly coordinating nodes to be seen as central, important players. Hossain suggests that individuals with higher network centrality have greater power within the social network, as they exhibit influence on a greater number of individuals and can control the flow of information between groups. The individual does not know the intentions of others beyond his neighbors, further accentuating the power relationships within the social network. These trends reveal a phenomenon known as the "collective sense", whereby the understanding of an individual reflects the shared knowledge forged from the development of this virtual culture. The creation of this virtual culture is a responsibility given to the most frequent communicators/coordinators, individuals that are not necessarily high-ranking employees. Hossain investigates this further by selecting a hierarchical list of 8 employee types ranging from "employee" to "CEO". From the dataset, he analyzes the average coordination score from each employee type, and finds that there is no correlation between hierarchical power and social networking power; that is, email-based coordination has little to do with organizational position. Hossain concludes that the results display a "qualitatively different form of governance" in which decisions are made more based on expertise rather than title.

Conclusions

Understanding organizational communication structures is important within a business environment, as it may reveal relationships about the levels of coordination,

participation, and influence each individual exerts. Hossain's findings show that network centrality is the key to understanding influence through a communication system. This research shows that technology is rapidly changing the governance structure of the hierarchical management system. It also suggests that new tools which integrate social network analysis and visualization directly into the communication subsystems would prove a valuable addition to future communication software.

Andrew Knight

Using planning poker for combining expert estimates in software projects (1/2)

Kjetil Moløkken-Østvold, Nils Christian Haugen and Hans Christian Benestad, Journal of Systems and Software, volume 81, pages 2106–2117, 2008

Background

The goal of the research was to analyze the effectiveness of the planning poker to combine estimates, using individual estimates for half of the tasks within the project as the control group.

Results

The research was done by comparing the statistical differences in the accuracy, optimism and composition in the tasks estimated with planning poker and the individually estimated tasks used as the control group, in addition the project members were individually interviewed.

The research and its results were divided into the following research questions:

Q1 “Are group consensus estimates less optimistic than the statistical combination of individual expert estimates?”

Q2 “Are group consensus estimates more accurate than the statistical combination of individual expert estimates?”

Q3 “Are group consensus estimates more accurate than the existing individual estimation method?”

Q4 “Does the introduction of a group technique for estimation affect other aspects of the developers’ work, when compared to the individual estimation method?”

Q4A “Are there differences between the planning poker tasks and the control group tasks that are related to the size of the changes?”

Q4B “Are there differences between the planning poker tasks and the control group tasks that are related to the complexity of the changes?”

In the research questions 1 and 2 were intended to compare intragroup differences while the later questions were intended to compare intergroup differences. Of the 55 tasks evaluated, 24 tasks were estimated with planning poker, 29 were used as the control group and 2 were removed from the dataset due to suspicion of faulty registration.

On the results for questions 1 and 2, based on the 24 planning poker tasks, it was found that the group discussion with the planning poker led to somewhat reduced optimism and a slight increase in accuracy compared to the statistical combination of the individual estimates. The results based on comparison of the 24 planning poker tasks and 29 control group tasks for question 3 showed similar accuracy on both the control group estimates and the estimates made with planning poker, both groups being on the average very accurate and unbiased. Of the 55 tasks 27 were found valid for question 4 and its sub-questions, and the results seemed to indicate that the control group’s tasks were larger while at the same time took less time than the planning

poker tasks, the larger amount of changes on the planning poker tasks compared to more deletions on the control group tasks were considered as a possible cause for the observed larger effort in the planning poker tasks. The larger effort in the planning poker tasks was also attributed to complexity which was theorized to be possibly caused by the way planning poker influences the planning of the task, or simply pure chance.

The interview goals were to gather background information on the project priorities, ask specific questions on the planning poker and to determine the interviewee's perceptions on the differences between planning poker and the individual estimations. Despite the statistical results the opinions on planning poker were generally positive and seemed to reinforce understanding and focus on the task at hand.

Conclusions

The research seemed to indicate that the use of planning poker could reduce optimism and sometimes slightly increase the accuracy of the estimations of the tasks. Planning poker was also suspected to increase the quality, or at least complexity, of the code. The interviews conducted seemed to indicate that the use of planning poker left a positive impression on the team and was assumed to increase the knowledge sharing and understanding of the project.

The research was conducted on a very small test group of only 53 tasks in a single software project, and contains a great deal of assumptions, theories and even statistical difference explained as coincidence. In the research it is also stated that there is very little empirical research on the subject and together with the small scale of the research leaves an impression that more research on planning poker as well as the other similar methods would be needed.

Matti Ollila

Using planning poker for combining expert estimates in software projects (2/2)

Kjetil Moløkken-Østvold, Nils Christian Haugen and Hans Christian Benestad, Journal of Systems and Software, volume 81, pages 2106–2117, 2008

Background

Planning poker is a tool for estimating task sizes in software projects. It is a tool for group planning, as the estimates are a combination of individual team members estimates. In planning poker each individual produces an estimate for a task, the estimates are shown and compared, and if the estimates are not the same, the result is discussed. Previous research has shown that combining many expert estimates reduces bias for optimism in producing estimates, but only on software projects. This is because unlike in other research, in software project research the people producing estimates were experts estimating real tasks that they had some familiarity with.

Results

The main research questions were whether group estimation or individual estimation produces more accurate results and whether the estimation method affects the way tasks are done. The experiment was done by using planning poker for half of the tasks in a software project, individual estimates for the other half. Also the code done on actual project was analyzed and people involved in estimation interviewed.

The study shows a slight decrease of optimism when discussing estimates. Interviews point that this is due to increased information about tasks. This resulted in group estimates being slightly more accurate than statistical combination of individual estimates. The accuracy was thus almost the same. The tasks randomly selected for group estimation ended up with both larger estimates and spent work hours than those selected for control group. In interviews it was found that the group found new subtasks and made some planning that resulted in more work hours spent on the tasks while discussing estimates.

Conclusions

Both styles of estimation produced almost equally good estimates, but with planning poker the complexity of the tasks was better taken into account. It seems that the benefit from planning poker is not more accurate estimates but better planned tasks that have input from different experts in the team. The sample size used for this study was quite small, but studies about task estimates are rare. Planning poker might be used for traditional task estimates, but this study was about using it in an agile project.

Juha Mäenalusta

Experience on knowledge-based software engineering: A logic-based requirements language and its industrial applications

Jeffrey J.P. Tsai, Alan Liu, Journal of Systems and Software, volume 82, pages 1578–158, 2009

Background

The article by Tsai and Liu discusses about knowledge-based software engineering (KBSE), formal requirements specification languages and their usage in industrial applications.

As an example of formal specification language, FRORL or Frame-and-Rule Oriented Requirements specification Language is described. FRORL was born in 1988 and its intent was to provide logically sound notation for describing implementation independent requirement specifications. Formal validation of the requirements, code generation based on the requirements and automation of software development in general were seen as possibilities with FRORL.

Results

FRORL, which is given as an example of formal language, uses frames for knowledge presentation. Frames consist of slots which can have values or actions attached to them. Rules can be used for specifying simple if-else type conditions for actions. Both static and dynamic requirements with time constraints can be specified with FRORL.

Automated processing applicable to FRORL specifications include verification of the specification and transformation to a procedural program. Verification step can include for example verification that all requirements are satisfiable and that timing constraints are applicable (requirement in real-time systems).

One interesting possibility represented by Tsai and Liu is the possibility to execute FRORL with an interpreter. This provides possibility to produce prototypes of the software in very early stage of the specification and also a way to debug the specification itself (instead of the implementation).

The article describes shortly five other systems similar to FRORL. The described systems can be used for specifying non-functional requirements, functional requirements or both in logical forms. Also some results on using natural language instead of logic based language is discussed. Usage of natural language would offer possibilities to employ more non-technical people in the software development process.

Tsai and Liu note that though formal specification languages still gather interest, the trend has slightly shifted towards model driven development (MDE). One example given by Tsai and Liu is Motorola's MouseTrap system which was heavily influenced by FRORL. MouseTrap was focused on providing specification methods for developing embedded real-time systems. MouseTrap enhanced the ideas FRORL had provided by using model driven development techniques. SDL and UML were used as a specification languages. Transformations were provided for generating code based on the formal specifications.

Results found from using model driven development show significant speed improvements in software development. In amount of code produced per person, a five time increase was seen, even though the used man-hours were decreased. Learning curve of new employees was also decreased with model driven development. Tsai and Liu write that the time required for training a new employee was decreased to 33-50 % of the original time. One important aspect mentioned is also that almost no defects were found in resulting code.

Conclusions

The article provided few examples of real world usage of formal specification languages. FRORL, Motorola's MousePad and results from model driven development were inspected more in depth. Some disadvantages were shortly brought up, but the main focus were on advantages on using formal specification languages, such as automated verification of the specification, code generation and more automated testing.

Olavi Karppi

An Introduction to agile methods

David Cohen, Mikael Lindvall and Patrica Costa, Advances in Computers, volume 62, pages 1-66, 2004

Background

Agile Methods are a reaction to traditional ways of developing software and acknowledge the "need for an alternative to documentation driven, heavyweight software development". The article introduces Agile methodology and analyzes whether or not Agile methods are useful in modern software development. The Agile Methods are actually a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

Results

The article analyzes the development models used in software development so far including Waterfall, Incremental, Spiral and Iterative. The main issues that have been focussed are tackling the changing requirements, managing risks using options like prioritizing requirements. The recent model which is one of the traditional methodology is Capability Maturity Model (CMM) and its successor CMMI. CMM mainly aims process consistency, predictability and reliability. Most Agile proponents argue that CMM concentrates more on documentation than code. While Agile proponents see a deep divide between Agile and traditional methods, some of the traditional model advocates think that Agile methods address many CMM practices.

A selection of Agile methods have been taken for examination - Extreme Programming (XP), Scrum, Crystal Methods, Feature Driven Development (FDD), Lean Development (LDD) and Dynamic Systems Development Methodology (DSDM). Each model is being analyzed where it works and where it is not. XP has its key principles as communication, simplicity, feedback, courage and quality work. Among all Agile models, XP has the shortest recommended iteration length, 2 weeks. It works with very minimum team size, from 2 to 10 and it does not have support for distributed teams.

Scrum is another most widely used Agile Method like XP, in which development process is unpredictable. Scrum resembles XP in many aspects. It has support for distributed teams unlike XP. The crystal method tries to address the poor communication problem. Crystal method allows any team size. FDD model is based on developing an overall model and building, planning, designing by feature. LDD is a management process rather than a development methodology, it doesn't directly address about team size or iteration length, etc., Dynamic Systems Development Method also doesn't address specifically about team size, iteration lengths, distribution because of its framework nature.

Modeling is a critical activity for project success. Agile Modeling addresses about how to model in an effective and Agile manner. It focuses on documentation and modeling and it can be used with any software development process. Each of the Agile methods have their own practices. For any organization, the successfulness in Agile adoption is based on its project and business characteristics. Volatile project requirements will fit into Agile methodology whereas very minimal percentage of

requirement changes will fit into traditional methods.

An eWorkshop conducted among Agile experts revealed various results. Most of them were matching with Agile proponents where some of them not. Many participants felt that any team could be Agile irrespective of size. Agile methods were criticized for their inability for the systems that have criticality, reliability and safety requirements. There was a consensus that Agile emphasis on testing, especially test driven development practice of XP, is the key to working with critical projects. The conclusion is that - experience with building systems is more important than experience in Agile Methods - Agile methods require less formal training than traditional methods - reliable and safety critical projects could be conducted using Agile - success factors are culture, people and communication - early warning signs - frequent refactoring and customer preferred documentation size.

The article also summarizes selected number of case studies on different aspects of using Agile methodologies. XP was used at an Online Telemarketing project, and all XP practices were introduced. The project was successful and they were managed to introduce planning game, collective ownership and customer on site successfully but it was observed that small releases and testing are difficult to introduce. In another case study, XP has been launched at a process intensive company. It has been observed from project management point of view that, the dependencies between features were almost non-existent since they followed the customer's priorities and not the priorities dictated by traditional design document.

Applying XP in maintenance environment revealed that pair programming and collection of metrics can improve process. When applying Scrum for the first time in an environment showed they were not fully prepared for Agile model. Based on a survey of using a XP/Agile across various organizations, the conclusion is as in the following: XP/Agile methods were working best on small projects, XP/Agile requires a cultural change with a well prepared workforce, focus should be on process and new practices need to be applied whenever required.

Conclusions

Agile methods can not necessarily be considered as the ones which are completely denouncing the traditional approach. Choosing whether being Agile or traditional approach is based on the application domain, criticality and innovativeness. Applications with highest level of criticality might have to go through traditional approach where as the same traditional approach may have negative impact on projects that need to be highly innovative and on the projects which are extremely sensitive to changes in market needs.

Saravanan Dhanabal

A practical use of key success factors to improve effectiveness of project management

Angela Clarke, International Journal of Project management, volume 17, issue 3, pages 139-145, 1999

Background

Management of a large project requires a large number of complex factors and activities for improving effectiveness. There are issues that need to be understood so that minimizing and elaborating these problems can be done. This is done by adopting Pareto's rule: "separate important few from trivial many". A study showed that there are four key factors in a project. Communication, clear objectives and scope, dividing project into bite-sized chunks and using project plans as working documents. Bite sizing is breaking a large project into sub-projects and it's regarded as one of the most important tasks in new or development projects.

Results

Results are based on observations of aerospace engineering companies and they can be adopted in most projects. The study presented seven problems which were solved using these key factors. Problems were: 1. Standardization can result in prescriptive project. Problem was solved by focusing on key factors in first stage and developing an auditing tool which is used in project. 2. Perceived return from project management is poor was solved by increasing awareness of perceived and actual benefit. This is done with better communication about benefits and results which can motivate people. Also focusing on important issues will decrease overload. 3. Project management is the same as project status reporter. Improving two-way information throughout the organization and making strategic information available to everyone can be a solution. 4. Commitment can disappear as project changes, especially in final stage. Communicating about the importance of the final stage can improve successfulness of the whole project. 5. Overload. Defining the 'bite sized chunks' at the outset of the project. This may allow a better estimate of resource requirements. 6. Cultural issues. Building individualism through effective communication. Encouraging of employee participation and redefining group boundaries can help with this problem. 7. Motivation. Project management seeing as a useless but a mandatory resource. This is solved with better communication to build more confidence. It may be done with by adopting a focused communication strategy.

Conclusions

Focusing on important parts can deliver great benefits and improve project management. It is recommended and useful that holistic approach to project management is taken, looking overall at project management and the main problems and the issues associated with its use. Applying Pareto principle can result in more effective management.

Aleksi Turpeinen

Reducing software product development time

John Callahan and Brian Moretton, International Journal of Project Management, volume 19, pages 59 – 70, 2001

Background

In this research paper authors are analyzing the impacts on the software product development time by various factors. The time it takes for the team to develop a software project is a deep concern of every project manager. The quest of making the development team more effective by decreasing the overall cost of the project and time spent on it involves analyzing many things from amount of testing you have in your project to the supplier involvement. The shorter the development time is, the faster your product arrives to the market, the greater your advantage will be over the competing products.

Results

In this paper authors are putting together a lot of research data on shortening the product development time. They mention a lot of other research papers and from these papers they form seven hypothesis:

1. More time spent planning is associated with shorter software development time.
2. More supplier involvement is associated with shorter software development time.
3. Sales and marketing involvement early in the development process is associated with shorter software development time.
4. Greater financial rewards for schedule attainment are associated with shorter software development time.
5. More frequent load builds are associated with shorter software development time.
6. Testing is associated with shorter software development time.
7. Greater project leader power is associated with shorter development time.

These hypothesis are then analyzed further by forming a special questionnaire and sending it to 107 Canadian and American software development companies. Fifty five product development project leaders of these companies agreed to complete the questionnaire. Questionnaires were sent out and returned via email. Forty four usable questionnaires were returned. The response rate for Canadian companies was 25/63, and for American companies, it was 19/44.

Callahan and Moretton defined a set of variables to analyze the questionnaire answers: planning, supplier involvement, sales and marketing involvement, schedule reward, build frequency, testing and project manager power. They also divided the results in groups by experience (in certain a project type) and project size (was measured by the cost of the project). The analysis has proven some of the assumptions to be correct for software development projects. However, these apply mostly to the less experienced teams that have not worked on a project of a certain type before. In the concluding word authors give some recommendations to less experienced teams such as: to involve suppliers early in the project development cycle (to have the latest tools available on the market) and to use frequent builds. There was also an advice to

involve sales and marketing department early in project development stages, this could give some crucial information on product requirements that is very valuable in the early stages.

Conclusions

This research paper gives some very valuable practical clues to the project managers. The further research is needed to analyze several aspects of the problem (such as the influence of the project leader), this paper also gives some leads to do that.

Mikhail Kapitonov

A project contingency framework based on uncertainty and its consequences

David Howell, Charlotta Windahl and Rainer Seidel, International Journal of Project Management, 2009, in press

Johdanto

Projektit ovat luonteeltaan erilaisia, joten niiden johtamisenkin on oltava erilaista. Projektinhallinta on yleisesti perustunut suunnitelmalliseen lähestymistapaan, mutta viime aikoina sen rinnalle on tullut ketteriä, kokemukseen perustuvia menetelmiä. Projektipäälliköillä ei kuitenkaan ole tietoa tai kokemusta näistä uusista menetelmistä. Projektin mahdollisuuksien malli (PCT) auttaa valitsemaan projektille sopivan hallintamenetelmän.

Tulokset

Tutkimus toteutettiin arvioimalla muiden tutkimusten tulosten soveltuvuutta PCT:hen. Katselmointiin valittiin 21 tutkimusta, joille oli yhteistä projektin ja organisaation riippuvuussuhteiden tarkastelu, joista oli viittauksia tai jotka toivat uusia näkökulmia aiheeseen. Tutkimuksissa toistuvia mahdollisuuksien teorian kanssa yhteneviä teemoja olivat projektin epävarmuus, kompleksisuus, kiireellisyys, ryhmän valtuuttaminen sekä kriittisyys. Teemoista kompleksisuus ja kiireellisyys voidaan nähdä projektin **epävarmuuden (U)** eri ilmentyminä, mikä on PCT -mallin ensimmäinen ulottuvuus. Sen sijaan kriittisyys ja ryhmän valtuuttaminen vaikuttavat projektin lopputulokseen, joten näistä saadaan PCT -mallin toiseksi ulottuvuudeksi epäonnistumisten **seuraukset (C)**.

Projektinhallinnan prosessit jaettiin niiden ominaisuuksien perusteella suunnitelmallisiin, ketteriin ja ongelmanratkaisumenetelmiin. Menetelmät sijoitettiin PCT -malliin niiden soveltuvuuden projektin epävarmuuden ja epäonnistumisen seurausten perusteella. **Suunnitelmalliset** menetelmät toimivat hyvin projekteissa, joissa epävarmuus on pientä, jolloin tulevaisuutta voi suunnitella tehokkaasti. **Ketterät** menetelmät ovat käytännöllisiä, kun epävarmuus on suuri ja epäonnistumisen seuraamukset pienet. **Ongelmanratkaisu** -menetelmät ovat parhaimmillaan korkean epävarmuuden ja epäonnistumisten seurausten projekteissa, joissa on tärkeää pienentää riskien todennäköisyyttä tai vähentää seurausten vakavuutta. Optimaalinen menetelmä voi myös vaihtua ajan myötä, kun projektin epävarmuus ja/tai seuraukset muuttuvat.

Yhteenveto

PCT -malli laajentaa yleistä mahdollisuuksien teoriaa projekteihin. Mallin hyödyllisyyden mittaa kuinka paljon siihen viitataan tulevissa tutkimuksissa. Tutkimus perustui kirjallisuuteen, joten seuraava askel on teorian kokeileminen käytännössä

Jaakko Palokangas

Does risk management contribute to IT project success? A meta-analysis of empirical evidence

Karel de Bakker, Albert Boonstra and Hans Wortmann, International Journal of Project Management, 2009, in press

Taustaa

Artikkelissaan *Does Risk Management Contribute to IT Project Success* de Bakker, Boonstra ja Wortmann käsittelevät riskienhallinnan merkitystä ohjelmistoprojektin lopputuloksen kannalta. He pyrkivät löytämään tähän kysymykseen vastauksen käymällä läpi 29 kappaletta eri IT – alan sähköisissä julkaisuissa julkaistua artikkelia vuosilta 1997 – 2009. Kriteerinä tutkimusaineiston valinnassa käytettiin sitä, että valittujen julkaisujen tuli sisältää nimenomaan empiiristä dataa riskienhallintaan liittyen.

Tulokset

Omien sanojensa mukaan de Bakker et al. tarkoituksena on myös jäsentää artikkelissaan riskienhallinnan ympärillä käynnissä olevaa keskustelua; he käyttävät esimerkkinä aiheesta ennen vuosituhannen vaihdetta esillä ollutta pelkoa siitä, että siirtyminen uudelle vuosituhannelle olisi rampauttanut tietojärjestelmiä ympäri maailman: ongelman ratkaisemiseen käytettiin suuret määrät rahaa, mutta mitään merkittävää ”Y2k-ongelmaa” ei koskaan esiintynyt, joka luonnollisesti herätti kysymyksiä satsausten merkityksellisyydestä. Tämä herätti heidän mukaansa myös keskustelua riskienhallinnasta yleisesti. Riskienhallintaa pidettiin tiettyssä mielessä välttämättömänä pahana, johon käytettiin aikaa vain siksi, että erilaisissa oppaissa niin käskettiin. Tähän keskusteluun Bakker et al. sanovat tuovansa uutta arvoa meta-analysoimalla tutkimissaan artikkeleissa esiintyviä empiirisiä todisteita riskienhallinnan merkityksestä projektin lopputulemaan.

Bakker et al. aloittavat aiheen käsittelyn erottamalla kaksi erillistä lähestymistapaa riskienhallintaan: evaluatiivisen sekä hallinnollisen tavan: Evaluatiiviselle lähestymistavalle tunnusomaista on pyrkimys tunnistaa ja listata tekijöitä, jotka voivat johtaa projektin epäonnistumiseen ja käyttää näitä tulevilla projekteilla riskien välttämiseen. Hallinnollisen tavan tyypillistä on puolestaan se, että selvitetään nimenomaan ne syyt, miksi projekti epäonnistuu, ja toimitaan näiden pohjalta. Näiden lisäksi käsitellään sitä, millä perusteilla ohjelmistoprojektin onnistumista voidaan mitata, ja lopulta tarkastellaan yksityiskohtaisemmin nimenomaan evaluatiivisen lähestymistavan merkitystä projektin onnistumiselle. Projektin onnistumisen mittareina käytetään kirjoittajien mukaan perinteisesti aikamääreitä, budjettia ja vaatimuskriteereitä siitäkin huolimatta, että tätä lähestymistapaa on usein kritisoitu. Nämä määritteet kirjoittajat ovat luoneet synteeseinä tarkastelun kohteina olleissa artikkeleissa esitetyistä ajatusmalleista; tällaisenaan niitä ei heidän mukaansa ole niissä esitetty.

Artikkelinsa tutkimustuloksia kuvaavassa osiossa de Bakker et al. toteavat, ettei kysymykseen riskienhallinnan vaikutuksesta ohjelmistoprojektin onnistumiseen voida vastata käyttämällä pelkästään evaluatiivista riskienhallintaan onnistumisen mittarina. Heidän mukaansa aiheesta kerätty empiirinen aineisto on yhä pääasiassa luonteeltaan aihetodisteiden kaltaista, eikä vastausta siksi ole saatavilla. Kirjoittajien mukaan jo se, että tiettyyn ohjelmistoprojektiin liittyvät eri asianomistajat voivat käyttää toisistaan

poikkeavia projektin onnistumisen mittareita johtaa siihen, että vaikutuksen yksiselitteinen määrittely ei onnistu.

Jesse Laanti

Use and benefits of tools for project risk management

T. Raz and E. Michael, International Journal of Project Management, volume 19, pages 9-17, 2001

Johdanto

Projektin johtamiseen liittyy monia tärkeitä elementtejä, joista riskien hallinta on yksi tärkeimmistä. Projektin johtamiseen ja myös riskien hallintaan on kehitetty useita eri työkaluja. Tutkimuksessa selvitettiin mitkä ovat yleisimmin käytössä olevia työkaluja ja miten hyvin ne tukevat riskien hallintaa.

Tulokset

Projektin- ja myös riskien hallintaan tarvitaan tehokkaita työkaluja. Nämä työkalut ovat usein kalliita, joten niiden osuus ja merkitys projektin kokonaiskustannuksissa voi olla merkittävä. On tärkeää, että osataan hankkia projektin riskienhallintaan parhaiten soveltuvat työkalut, jotta saadaan sijoitukselle vastinetta. Tutkimus keskittyi vertailemaan suosituimpia projektin riskienhallinnan työkaluja ja siten löytämään niistä parhaiten riskienhallintaa tukevat. Työkaluilla tässä tutkimuksessa tarkoitetaan varsinaisten sovellusten lisäksi myös käytäntöjä ja prosesseja, joilla riskejä hallitaan.

Tutkimus suoritettiin kyselyllä, joka lähetettiin postin tai sähköpostin välityksellä noin 400 projektipäällikölle ohjelmisto- ja korkean teknologian yrityksissä vuonna 1998. Vastauksista käyttökelpoisia oli 84 kpl, jotka otettiin mukaan tutkimukseen. Kysely oli jaettu kolmeen pääalueeseen, joista jokainen sisälsi joukon kysymyksiä, joihin vastattiin asteikolla 0-5. Ensimmäisessä osiossa selvitettiin, kuinka suuri vaikutus yksittäisellä työkalulla on projektin onnistumiseen. Tätä varten oli työryhmän kanssa, joka koostui viiden eri yrityksen projektipäälliköistä, suodatettu yli 100 työkalusta 38, jotka toimivat tutkimuksen pohjana. Toinen osio selvitti, kuinka tehokasta ja vaivatonta projektin johtaminen ylipäättään organisaatiossa oli sillä toimintatavalla, joka kyseisessä organisaatiossa oli käytössä. Kolmannessa osiossa selvitettiin kuinka suuri osuus riskienhallinnalla on projektin onnistumisen kannalta.

Yksittäisen työkalun vaikutus riskien hallinnassa oli vastausten mukaan suurin projektin vaatimusten- ja alihankkijoiden johtamisessa sekä kokoonpanon kontrolloinnissa eli toisin sanoen projektin taustatekijöiden hallinnassa. Kehittämistyön menetelmistä simulointi ja prototyyppien tekeminen havaittiin myös tärkeiksi tekijöiksi yksittäisen työkalun suhteen. Yksittäisen työkalun vaikutusta ei sen sijaan koettu merkityksellisenä esimerkiksi riskien kontrolloinnissa.

Vastaukset projektin johtamisen vaivattomuudesta ja tehokkuudesta yhdistettiin ensimmäisestä osiosta saatuihin vastauksiin, jolloin nähtiin korreloiko työkalun kokonaisarvosana projektin johtamisen kanssa. Tulos oli, että korrelaatio oli varsin selvä eli työkaluilla näytti todella olevan merkitystä projektien johtamisen kannalta.

Kolmannen osion tuloksia verrattiin ensimmäisen osion tuloksiin, jotta saataisiin selville onko yksittäisen työkalun merkityksellä vaikutusta riskien hallinnan merkitykseen. Ne projektipäälliköt, jotka pitivät riskien hallinnan merkitystä projektin onnistumisen kannalta vähäisenä, olivat myös antaneet selkeästi vähemmän painoarvoa yksittäisen työkalun merkitykselle riskien hallinnassa.

Johtopäätökset

Tutkimuksen avulla nousi esiin työkaluja, joiden voidaan todeta hyödyttävän projektin riskienhallinnassa. Tuloksia voi käyttää hyväksi, esim. vertailemalla millä riskienhallinnan osa-alueilla omassa organisaatiossa on parantamisen varaa ja miettiä tukisiko jokin kyseisellä osa-alueella hyväksi havaituista työkaluista riskienhallintaa paremmin. Tavoitteena oli tutkia, mitä työkaluja sellaiset yritykset käyttävät riskienhallintaan, jotka ovat selkeästi siinä menestyneet. Tutkimus onnistui tuomaan esille nämä työkalut eli sitä voidaan pitää onnistuneena.

Antti Jämsén

Improving project outcomes through operational reliability: A conceptual model

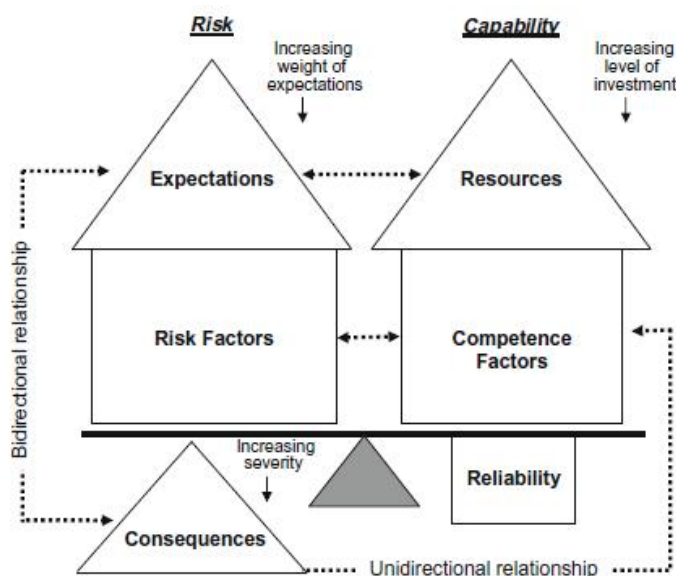
John Sullivan and Roger Beach, International Journal of Project Management, volume 27, issue 8, 765–775, 2009

Johdanto

Artikkelissaan Sullivan ja Beach käsittelevät HRO-organisaatioiden (high reliability organization) monimutkaisiin järjestelmiin liittyvien projektien onnistuneeseen läpivientiin liittyviä tekijöitä. Nämä organisaatiot erottuvat muista siinä, että ne asettavat luotettavuuden ja turvallisuuden tärkeimmiksi prioriteeteikseen, perinteisten tuottavuuden ja tehokkuuden sijaan. Nämä organisaatiot ovat tyypillisesti sellaisia, joiden toiminnassa tapahtuvat virheet voisivat aiheuttaa suurta tai näkyvää vahinkoa, kuten esimerkiksi ydinvoimalat.

Tulokset

Artikkelin mallissa (katso Kuva 1) organisaation toimintojen luotettavuuteen esitetään tekijöitä (odotukset, riskit, resurssit, kyvyt ja seuraukset) jotka tulee huomioida, jotta monimutkainen järjestelmä saadaan kehitettyä ja hallittua onnistuneesti. HRO-organisaatioiden projektien luonteeseen kuuluu osata pitää nämä tasapainossa. Tällöin organisaatiolla on riittävät resurssit toiminnan riskeihin nähden ja järjestelmät jatkavat toimintaansa luotettavasti. Hyvät lähtökohdat luovat korkeat odotukset erityisesti toiminnan turvallisuudelle, mutta vastaavasti vähenevä osaaminen ja resurssit tulisi näkyä laskevana odotuksena projektien ja muiden toimintojen tuloksissa, jottei korkea luotettavuus kärsisi.



Kuva 1 Sullivanin ja Beachin malli toiminnallisesta luotettavuudesta.

Kasvavista odotuksista seuraa myös uhka odotusten pettämisestä ja sitä seuraavasta reaktiosta jolloin organisaation on entistä vaikeampi saada hyväksyntää toiminnalleen. Artikkelissa mainitaan tärkeäksi myös oppiminen epäonnistuneista projekteista, kuten Nasa oli tehnyt Apollo 13:n vastoinkäymisten jälkeen. HRO-organisaation piirteisiin

kuuluukin jatkuva halu oppia. Tärkeätä on myös olla luottamatta liiaksi omaan osaamiseen ja ylläpitää osaamistaan. Nasa ei onnistunut tässä, joka myöhemmin kostautui Challenger sukulan tuhoutuessa. Kirjoittajat mainitsevatkin Nasan ongelmana olleen tässä ns. groupthink, jossa ryhmässä muodostuu paineita luoda yksi yhtenäinen mielipide ja sitä tavoiteltaessa onnistumisen kannalta tärkeä kysymys saatetaan sivuuttaa.

Artikkelissa kerrottiin kahden eri HRO-organisaatioiksi miellellävän haastattelusta, joilla mallia testattiin käytännössä. Lisäksi mallia testattiin erilaisista onnettomuuksista julkaistuihin tietoihin. Esimerkkitapauksena käytetään Nasan sukulan tuhoutumista, johon johtivat jo pitempään jatkunut resurssien karsiminen ja osaamisesta huolehtimatta jättäminen. Tästä huolimatta organisaation toiminnan odotettiin jatkuvan entisellään. Kuten artikkelissa esitettiin, Nasalta odotettiin enemmän vähemmällä. Toisena esimerkkinä on Yhdysvalloissa tapahtunut laaja sähkökatko, jonka jäljiltä sähkönjakelun palautuminen normaaliksi kesti huomattavan pitkään. Katkoon johtivat pitkät laiminlyönnit järjestelmän ylläpidossa ja kyvyttömyys oppia aiemmin ilmenneistä pienemmistä ongelmista. Näiden testien perusteella mallin todettiin vastaavan todellisuutta.

Johtopäätökset

Sullivan ja Beach toteavat, että monimutkaisten tietojärjestelmäprojektien toteutuksen ja näiden järjestelmien ylläpidon onnistuminen on riippuvaista oikeanlaisesta yrityskulttuurista, joka HRO-organisaatioissa on omaksuttu. Kun tämä kulttuuri ymmärrettäiseen paremmin, sen piirteet voitaisiin siirtää tavallisiin organisaatioihin, jolloin niiden toimintaa saataisiin luotettavammaksi. Mallia soveltamalla jo ennen projektien alkamista saataisiinkin ongelmia ratkaistua jo ennen niiden syntymistä ja organisaatioiden toimintaa entistä luotettavammaksi.

Timo Taipalus

Global virtual teams for value creation and project success: A case study

Liz Lee-Kelley and Tim Sankey, International Journal of Project Management, volume 26, pages 51-62, 2008

Johdanto

Tutkimuksen tavoitteena oli selvittää, eroavatko virtuaalisten projektien haasteet perinteisistä projekteista ja voivatko virtuaaliset projektit joissain tilanteissa olla perinteisiä projekteja tehokkaampia. Tutkimuskohteina käytettiin sisäisen toimintamallinsa uudistaneen kansainvälisen pankin kahta IT-projektia, joiden työntekijät olivat eri puolilta maailmaa.

Avainsanoja: Virtuaalinen projekti, kansainvälinen virtuaalinen tiimityöskentely, etätyö

Tulokset

Tutkimus toteutettiin haastatteluihin perustuvana tapaustutkimuksena. Tutkittavissa tiimeissä työskenteli satoja henkilöitä, mutta haastattelut suunnattiin vain projektien johtajille. Haastatteluita tehtiin yhteensä yksitoista, ne toteutettiin puhelimitse ja keskustelut nauhoitettiin. Analyysivaiheessa keskityttiin selvittämään kansainvälisten virtuaalitiimien rakennetta, projektien onnistumista ja onnistumisen mittaamista sekä ongelmia ja niiden ratkaisuja.

Tutkimuksesta selvisi, että tapaustutkimuksen kohteena olleet projektit olivat onnistuneet sekä ajallisesti että taloudellisesti. Tilaaajien ja osakkeen omistajien tyytyväisyyttä ei kyetty mittaamaan, mutta muilta osin tutkimusta ja sen tuloksia pidettiin onnistuneina. Selvisi, että pienistä kulttuurillisista ja aikavyöhykkeiden aiheuttamista eroista huolimatta virtuaalinen yhteistyö tiimeissä toimi hyvin. Projektien jäsenillä ei ollut huomattavia vaikeuksia sopeutua virtuaalityöskentelyyn.

Tutkimuksen johtopäätöksissä ehdotetaan, että vastaavia projekteja suunniteltaessa olisi hyvä sisällyttää työryhmiin henkilöitä, joilla on aiempaa kokemusta kansainvälisestä virtuaalisesta tiimityöskentelystä. He voivat ohjata toisia tiimin jäseniä tarvittaessa. Myös tiimin jäsenten halukkuutta kommunikoida ja kehittyä tulisi rohkaista.

Yhteenveto

Tutkimus osoitti, että kansainvälisen virtuaalisen tiimityöskentelun avulla voi toteuttaa projektin onnistuneesti. Virtuaalisten tiimien työskentelyssä esiin nousseet ongelmat ovat yleisiä myös perinteisten tiimien toiminnassa. Virtuaalinen ryhmätyöskentelymuoto voi olla ihmisille vielä hieman oudoksuttava, mutta tutkimustulosten valossa sen voi nähdä yleistyvän jatkossa.

Heidi Vaarala

Managing user expectations on software projects: Lessons from the trenches

Stacie Petter, International Journal of Project Management volume 26, issue 7, pages 700-712, 2008

Johdanto

Monet aikaisemmat tutkimukset ovat osoittaneet, että käyttäjien vääristyneillä odotuksilla on suora negatiivinen vaikutus heidän tyytyväisyyteensä ohjelmistoprojektin lopputta. Käyttäjille syntyy aina ohjelmistoprojektin aikana oletuksia siitä miten uusi järjestelmä tulee toimimaan. Käyttäjille ohjelmiston laatu määräytyy suurelta osin siitä miten hyvin järjestelmä täyttää nämä odotukset, vaikka ne eivät olisi millään tavalla realistisia tai edes liity järjestelmän oikeaan toimintaan. Tämän vuoksi on tärkeää saada käyttäjien odotukset pysymään realistisina ja oikeassa aiheessa. Kuitenkaan tutkimuksia siitä kuinka käyttäjien odotuksia tulisi hallita ei ole juurikaan tehty. Tässä tutkimuksessa keskityttiin löytämään keinoja joilla käyttäjien odotukset saataisiin pysymään projektinjohdon hallussa.

Tulokset

Tutkimus suoritettiin haastatteleamalla kahtatoista projektinjohtajaa suuresta IT-johtamis- ja konsultointiyhtiöstä. Haastateltavia pyydettiin kertomaan kaksi kokemaansa käytännön tapausta jossa he ovat joutuneet tekemisiin käyttäjien odotusten hallinnan kanssa, yksi onnistunut ja yksi ei-niin-onnistunut tapaus. Lisäksi haastateltavat saivat kertoa muita kokemuksen tuomia ohjeita odotusten hallintaan liittyen.

Saadut vastaukset koottiin kahteen taulukkoon, onnistuneet ja ei-niin-onnistuneet, joiden avulla vertailtiin näissä käytettyjä taktiikoita. Taulukoista tunnistetut taktiikat kerättiin toisaalle ja eroteltiin onnistuneisiin ja epäonnistuneisiin. Osa taktiikoista tunnistettiin myös projektinjohtajien jälkeinpäin ajatellen keksimistä "näin olisi pitänyt tehdä" oivalluksista. Löydetyt taktiikat eroteltiin loogisiin ryhmiin ja tämän jälkeen käytiin kaikki haastatteluiden tapauksen uudelleen läpi jotta voitiin varmistua, että onnistuneissa tapauksissa käytettiin onnistuneiksi luokiteltuja taktiikoita ja toisinpäin.

Taktiikat jakaantuivat kolmeen laajempaan strategiaan: käyttäjien osallistuminen, johtaminen ja luottamus.

Käyttäjien osallistuminen oli kaikkein yleisin strategioista. Työskentelyn tulisi tapahtua käyttäjien kanssa eikä ainoastaan käyttäjille. Heitä tulee kuunnella ja pitää heidät ajantasalla. Lisäksi käyttäjien tulee antaa päättää tärkeimmistä asioista budjetin, aikataulun ja toiminnallisuuden suhteen, mutta heille pitää tehdä selväksi mihin ja miten heidän päätöksensä projektissa vaikuttavat. Yleisin syy epäonnistumiseen käyttäjien odotusten hallinnassa on jättää käyttäjät ulos projektin kehityksestä.

Johtamista tarvitaan kahdenlaista: käyttäjien ja projektiryhmän johtamista. Käyttäjien keskuudesta tarvitaan joku joka tuntee projektin tavoitteet hyvin ja pystyy kertomaan projektin hyödyistä muille käyttäjille ja osakkaille. Projektiryhmä voidaan kokea ulkopuoliseksi ja tällainen, tutkimuksessa projektin esitaistelijaksi nimetty, henkilö on paljon tehokkaampi käyttäjien ja projektin välisessä vuorovaikutuksessa.

Projektiryhmän johto keskittyy siihen, että projektiryhmä saa tulosta aikaiseksi, mutta ei kuitenkaan saa unohtaa käyttäjiä ja myös heidän kantiltaan asioiden ajattelemista.

Luottamus täytyy myös olla kunnossa projektiryhmän ja asiakkaan välillä. Asiakkaalle pitää pystyä kertomaan myös huonot uutiset ja se, jos jokin ei ole mahdollista toteuttaa annettujen resurssien puitteissa. Asiakkaan tulisi kuulla ongelmista projektinjohdolta heti eikä myöhemmin jostain muualta. Näin asiakas ei muodosta odotuksia asioille joita projekti ei pysty täyttämään. Jos luottamus asiakkaan välillä menetetään, se on hyvin vaikeaa enää saada takaisin ja tämä saattaa olla erittäin merkityksellistä projektin jatkamisen kannalta.

Yhteenveto

Tutkimus antaa itselleen kritiikkiä ja osoittaa ettei se ole kaiken kattava. Ensinnäkin se perustuu kapeaan otantaan konsultointiyhtiön projektinjohtajista. Toisekseen suurin osa tarkastelluista projekteista liittyivät valtiollisiin organisaatioihin. Lisäksi osa haastatelluista projektinjohtajista omasi vain kolmen vuoden kokemuksen työstään, vain yksi haastatelluista omasi laajan kahdenkymmenen vuoden kokemuksen, loppuilla oli viidestä seitsemään vuotta kokemusta työstään. Myöskään kahdentoista hengen otanta ei ole tilastollisesti kovin vertailukelpoinen. Näin ollen tutkimuksen tuloksia ei voi pitää kaiken kattavana listana parhaista strategioista, vaan ennemminkin hyvänä pohjana lisätutkimuksille aiheesta.

Tutkimus on silti hyödyllinen projektien johtajille, sillä se tarjoaa, ainakin jossain määrin onnistuneita, käytännön keinoja käyttäjien odotusten hallintaan ja täten voi auttaa parantamaan ohjelmistoprojektien laatua käyttäjien silmissä. Vaikka projekti olisi osittain puutteellinen, voivat käyttäjät silti olla siihen tyytyväisiä ja käyttäjän tyytyväisyys on ohjelmistoprojektien yksi tärkeimmistä ellei tärkein tavoite.

Matti Pesonen

The interactive effect of team dynamics and organizational support on ICT project success

Roy Gelbard and Abraham Carmeli, *International Journal of Project Management*, volume 27, pages 464-470, 2009

Johdanto

Tutkimuksessa pyrittiin empiirisellä testaamisella havainnollistamaan informaatio- ja kommunikaatioteknologisten projektien ryhmädynamiikan (tiedon jako, osallistuminen, kollektiivisuus ja toisilta oppiminen) ja organisaation tuen välistä monimutkaista vuorovaikutusta. Tutkittavia osa-alueita olivat budjetit, yhteistoiminta ja ajankäyttö.

Tulokset

Projektin onnistuminen muodostuu kyvystä tuottaa asiakasta tyydyttäviä tuotteita, antaa tiimin jäsenille mahdollisuus oppia uutta sekä parantaa tiimin kykyjä suoriutua tulevista tehtävistä. Tältä pohjalta tutkimuksessa testattiin kolme hypoteesia:

1. Projektin menestys riippuu tiimistä, ja sen toiminta sisäisestä ryhmädynamiikasta.
2. Organisaation tuki vaikuttaa positiivisesti projektin menestykseen.
3. Organisaation tuki muokkaa ryhmädynamiikan ja projektin onnistumisen suhdetta. Positiivinen vaikutus on suurempi tiimeillä, jotka saavat laajempaa tukea.

Tutkimus toteutettiin alkuvuodesta 2006 kyselytutkimuksena 254 alan projektipäällikölle. Heistä kyselyyn vastasi 191. Kysymyksiä oli 250 kappaletta jaoteltuina kolmeen ryhmään: budjetti ja ajankäyttö, yhteistoiminta tiimin jäsenten kesken ja projektipäälliköiden havainnot organisaation tuesta. Vastausvaihtoehdot annettiin viisiportaisen Likert-asteikon muodossa. Kontrollimuuttujina käytettiin uravuosia organisaatiossa ja projektin johtotehtävissä sekä sukupuolta.

Kyselyn tuloksista laskettiin tilastolliset tunnusluvut. Nämä tukivat esitettyjä hypoteeseja siten, että ryhmädynamiikka oli sujuvampaa laajaa organisaation tukea nauttivissa projekteissa. Tämän todettiin vaikuttavan positiivisesti myös projektin menestystä mittaaviin tekijöihin.

Yhteenveto

Ryhmädynamiikka määrittää suurelta osin projektin menestyksen. Organisaatiot ovat avainasemassa mahdollistamaan olosuhteet tällaisen tiimityöskentelyn synnylle. Tutkimuksen valossa niiden kannattaa tähän panostaa parantaakseen projektiansa menestystä.

Ari Koivuniemi

Top management support: Mantra or necessity?

Raymond Young and Ernest Jordan, International Journal of Project Management, volume 26, pages 713-725, 2008

Johdanto

Raymond Youngin ja Ernest Jordanin kirjoittama artikkeli "Top Management Support: Mantra or Necessity?", joka julkaistiin International Journal of Project Management -lehdessä, pohtii nimensä mukaisesti sitä, onko korkean johdon tuki (top management support = TMS) tarpeellista IS projekteille vai ei. Artikkelitoteaa, että tällä hetkellä puuttuu näyttö siitä, että TMS olisi tarpeellista projektille ja olisi sopimatonta muuttaa käytäntöjä ilman vahvaa näyttöä sen tarpeellisuudesta. Artikkelin tarkoituksena on siis saada näyttöä siitä, kuinka tärkeää TMS on ja nimenomaan korkean johdon ja johtokunnan näkökulmasta.

Tulokset

Artikkelin raportoima tutkimus suoritettiin kuvailevalla tapaustutkimuksella (descriptive case study). Maksimaalinen tiedonsaanti varmistettiin valitsemalla tavallisia ja esimerkillisiä organisaatioita. Organisaatiot valitsi työryhmä, jossa oli edustajia 49 yrityksestä. Artikkelin mukaan TMS on jatkuvasti todettu olevan tärkeä, joten ei pitäisi olla väliä, minkälaisia yrityksiä tutkimukseen valittiin. Tutkimus suoritettiin reilun kahden vuoden aikana, aina yksi tapaus kerrallaan. Tutkimusaineistoa kerättiin monella eritavalla: haastatteluilla, projektin dokumentaatiota tutkimalla sekä tarkkailemalla.

Tutkimuksen aikana eri yritysten projektit pärjäivät eri tavoin. Yksi projekti epäonnistui (TechServ), yksi menestyi todella hienosta (SkyHigh) ja loput kolme (TechMedia, ABS ja Agency) siltä väliltä. TechServ:in epäonnistuminen johtui kahdesta asiasta: Ensinnäkin siitä, että vain osa projektin tavoitteista saatiin suoritettua. Toiseksi siitä, että projektista johtuen organisaation liikevaihto väheni 25 % ja se menetti maineensa asiakkaiden silmissä.

SkyHigh:n menestys johtui myös monen asian summasta. Artikkelin mukaan näitä asioita olivat: projekti pysyi aikataulussa ja budjetissa, sen korkea johto oli hyvin perillä projektin tavoitteista ja kulloisestakin edistymisestä, sekä se oli heti mukana ratkaisemassa vastaan tulleita ongelmia.

TechMedia menestyi vain osittain sillä sen tulokset eivät yltäneet sille tasolle, joka arvioitiin projektin aikana, mikä johti pettymykseen projektin päätyttyä. ABS koki menetyksen kesken projektin, kun sen sponsori lähti, joka aiheutti sen, ettei projektia viety niin pitkälle kuin se olisi ollut mahdollista, jos sillä olisi ollut sponsori. Ja Agencyä vaivasi nuori projektitiimi, joka kylläkin sai projektin loppuun asti, pysyi budjetissa ja aikataulussa, muttei pystynyt kuitenkaan ihan niin hienoon lopputulokseen, että projekti olisi voitu lukea täydelliseksi menestykseksi.

Johtopäätökset

Artikkelissa määritellään viisi kriittistä menestyksen tekijää (critical success factor = CSF): TMS, käyttäjän osallistuminen, projektin metodologia, korkean tason suunnittelu ja projektin henkilöstö. Jokainen näistä kohdista käydään yksitellen läpi verraten projekteja toisiinsa ja tullaan siihen tulokseen, että TMS on näistä kaikkein

tärkein CSF. Millään muista CSF:sistä ei ole samanlaista valtaa, kuin korkean johdon tuella, saattaa projekti loppuun menestyksekkäästi. Artikkelissa todetaan lopuksi, että korkean johdon täytyy tulevaisuudessa hyväksyä se, että heillä on henkilökohtaisesti suurin vaikutusvalta siihen tuleeko projekti menestymään vai epäonnistumaan.

Jenni Haataja

Why projects fail – How contingency theory can provide new insights? A comparative analysis of NASA's Mars Climate Orbiter loss

Brian J. Sauser, Richard R. Reilly and Aaron J. Shenhar, International Journal of Project Management, volume 27, pages 665-679, 2009

Johdanto

NASA:n Mars Climate Orbiterin (MCO) tarkoituksena oli kerätä tietoja Marsin ilmastosta. Tälle projektille oli varattu todella pieni budjetti sekä hyvin tiukka aikataulu, jonka takia MCO:n vaatimukset jäädytettiin jo aikaisessa vaiheessa. Rahan ja ajan säästämiseksi käytettiin alijärjestelmiä edellisistä projekteista, eikä kaikkia kriittisiä järjestelmiä testattu uudestaan. Täydelliseen katselmointiprosessiin ei myöskään resurssit riittäneet. MCO onnistuttiin laukaisemaan ajallaan ja sen lento Marsin sujui hyvin, mutta hieman sen jälkeen, kun se aloitti toimenpiteet Marssin kiertoradalle pääsemiseksi, avaruusaluksen signaali katosi, eikä sitä pystytty enää paikantamaan.

Tulokset

MCO:n epäonnistumista tutkittiin käyttämällä kolmea erilaista projektinjohtamisen kontingenssiteorian kehystä, jotka olivat Hendersonin ja Clarkin puitteet analysoida innovaatioita, Shenharin ja Dvirin NTCP kontingenssipuitteet projektin luokittelulle ja Pichin, Lochin ja de Meyerin selviytymisstrategiamalli epävarmuudelle.

Hendersonin ja Clark ajattelevat tuotteen koostuvan komponenteista ja niiden välisistä liitännöistä ja että erilaisilla teknologisilla muutoksilla on erilaisia vaikutuksia. Heidän kehyksessään on neljä yksittäistä ulottuvuutta. Radikaali uudistus vaikuttaa sekä komponentteihin että liitännöihin niiden välillä. Inkrementaalilla uudistuksella ei ole juurikaan vaikutusta komponentteihin eikä niiden välisiin liitännöihin. Modulaarinen uudistus vaikuttaa vain komponentteihin. Arkkitehtuurinen uudistus vaikuttaa vain tuotteen liitännöihin.

Shenharin ja Dvirin kehyksen tarkoituksena on löytää projektille optimaalisin hallintatyyli. Kehys koostuu neljästä ulottuvuudesta, jotka ovat uutuus (novelty), monimutkaisuus (complexity), teknologia (technology) ja vauhti (pace). Uutuudessa yritetään kartoittaa sitä, miten uusi tuote on asiakkaille. Monimutkaisuudessa yritetään löytää tuotteen paikka järjestelmien hierarkiassa. Teknologiassa taas selvitetään missä suhteessa uutta teknologiaa on käytetty ja vauhti kertoo kuinka kiireellinen tuote on.

Pichin, Lochin ja de Meyerin kehyksen tarkoituksena on tarjota erilaisia strategioita projektin hallintaa, kun projektissa tarvittavien tietojen soveltuvuudesta ei olla varmoja. Strategioita on neljä. Oppimisstrategiassa tarvitaan oppimista erilaisista asioista, jotta päästään optimaaliseen ratkaisuun. Ohjaavassa strategiassa ei tarvita oppimista päästäkseen optimaaliseen ratkaisuun. Oppimisessa ja valitsemisessa valitaan se projekti, joka etenee parhaiten ja oppiminen tapahtuu eri projektien välisessä tiedon vaihdossa. Valitsemisstrategiassa ei tarvita oppimista, vaan valitaan parhaiten sopiva projekti kyseiseen tilanteeseen.

Hendersonin ja Clarkin kehyksellä luokiteltuna MCO oli akkritehtuurinen uudistus, joka on vaikeasti tunnistettavissa. Tähän tunnistamiseen menee yleensä paljon aikaa, mitä MCO:lla ei ollut rajallisen budjetin takia. Projektiin yritettiin myös käyttää uutta ”nopeammin, paremmin ja halvemmalla” –menetelmää. NASA ei kuitenkaan tarjonnut tarpeeksi ohjausta, miten tällaista menetelmää tulisi käyttää.

Shendarin ja Dvirin kehyksen avulla testattiin, oliko MCO:n projektityyppi yhdenmukainen projektin hallinnan lähestymistavan kanssa. Sen perusteella tuli selväksi, että projektia johdettiin oikein ainoastaan projektin vauhdin perusteella. Uutuuden, teknologian ja monimutkaisuuden ulottuvuudet aliarvioitiin.

Pichin, Lochin ja de Meyerin kehyksen avulla arvioituna MCO:hon olisi pitänyt käyttää oppimisstrategiaa, mutta sen sijaan käytettiin ohjaavaa strategiaa. Oppimistrategiaa olisi puoltanut se, että projektissa oli paljon kokemattomia ihmisiä, jotka olisivat tarvinneet koulutusta.

Johtopäätökset

Kehykset osoittivat, että MCO:n epäonnistuminen johtui johtamisen epäonnistumisesta ja syitä tähän oli monia. Projektiin yritettiin käyttää uutta menetelmää, joka ei toiminut tämän kaltaisessa projektissa. Projektin arkkitehtuurin uudistus jätettiin huomioimatta. Sitä johdettiin väärin kolmella eri tavalla: uutuuden, teknologian ja monimutkaisuuden suhteen. Ohjaavastrategia oli myöskin väärä valinta, sillä osa projektihenkilöstöstä olisi tarvinnut koulutusta.

Jotkin osat artikkelista olivat epäselviä ja vaikeasti ymmärrettäviä kuten Hendersonin ja Clarkin kehys sekä Pichin, Lochin ja de Meyerin kehys. Näitä kehyksiä ei selitetty tarpeeksi ymmärrettävästi ja käytetty kieli oli vaikeasti luettavaa.

Niina Ojala

Software project management anti-patterns

Ioannis Stamelos, Journal of Systems and Software, 2009, in press

Johdanto

Artikkeli käsittelee ohjelmistoprojektin johtamisen antimalleja (*eng. anti-pattern*) – yleisiä ongelmatilanteita – ja niiden seurauksia. Johtamisen antimallilla voidaan tarkoittaa huonoa käytäntöä, vääränlaista reagointia tiettyihin tapahtumiin tai minkä tahansa tekijän epäonnistunutta ymmärtämistä, hallintaa tai ennustamista projektissa. Artikkelissa luodaan katsaus tähän asti raportoituihin ja dokumentoituihin antimalleihin ja painotetaan tarvetta työkaluille, joilla antimalleja voitaisiin formaalisti esittää. Tekstissä esitellään kolme olemassa olevaa, erityyppistä formalismia ja kuvataan kuinka työkaluilla voidaan tarjota puitteet antimallien tunnistamiselle, dokumentoinnille ja levittämiselle.

Tulokset

Tutkimalla kirjallisuutta on havaittu, että kirjallisuutta ohjelmistoprojektin johtamisen hyväksi todetuista käytännöistä ja suunnittelumalleista on olemassa, mutta huonot käytännöt ja antimallit ovat jääneet vähemmälle huomiolle. Toinen havainto on, että tähän asti raportoidut antimallit ovat esitetty epämuodollisesti, usein ainoastaan tekstiselityksenä. Eri antimallien on havaittu olevan usein yhteydessä toisiinsa ja epämuodolliset esitystavat eivät pysty osoittamaan kausaalisuus- tai muitakaan suhteita eri antimallien välillä. Epämuodolliset esitystavat ovat myös jarruna asiantuntijoiden väliselle kommunikoinnille ja hidastavat antimallien tuntemuksen leviämistä.

Tähän asti ehdotetuista formalismeista esitellään kaksi. Ensimmäisenä kuvataan Bayesian Belief Networks – asykliset graafit, joiden etu on todennäköisyyksiin perustuvien syy-seuraussuhteiden havainnollistaminen.

Toisena esitellään ontologiat, jotka havainnollistavat hyvin antimallien yleisen ja yhteisen käsitteistöön ja rakenteen. Ontologioiden etuna on myös yhtymäkohdat ohjelmistoarkkitehtuurien tai ohjelmistokehityksen antimallien kuvauksen kanssa, jolloin niiden voidaan olettaa tavoittavan suuren yleisön helpommin.

Antimallien esityksen formalisoinnista seuraa automaattisesti myös mahdollisuus työkalujen kehitykselle antimallien hallintaan. Internetissä, mahdollisesti projektipäälliköiden yhteisöissä toimivilla työkaluilla voitaisiin edistää antimallituntemuksen leviämistä ja tietämyksen siirtymistä teoriasta käytäntöön. Työkaluista mainitaan Thessalonikin yliopistossa kehitteillä oleva työkalu, jolla on tarkoitus auttaa projektipäälliköitä tunnistamaan ja hyödyntämään antimalleja. Järjestelmä osaa listata antimallien mahdollisia oireita, syitä ja seurauksia. Lisäksi käyttäjä voi syöttää järjestelmään uusia antimalleja. Web-käyttöliittymä tarjoaa projektipäällikölle mahdollisuuden eri tyyppisiin kysymyksiin. Esim. ”Olen havainnut projektissani ei-toivotun tilanteen X, mikä voisi olla sen syy ja miksi se on tapahtunut?”

Tämänkaltaisten työkalujen arvellaan olevan hyödyllisiä kaventamaan yleisesti tunnistettua ohjelmistoprojektin johtamisen teorian ja käytännön välistä kuilua.

Yhteenveto

Artikkeli perustelee antimallien tuntemuksen hyödyllisyyden ohjelmistoprojektien hallinnassa, osoittaa alueen vallitsevat ongelmat ja esittää niille ratkaisuja. Erityisesti painotetaan työkalujen kehittämisen tarvetta, jolloin antimallitietämyksen ylläpito, tutkimus ja käytäntöön soveltaminen helpottuu. Antimallit tarjoavat yhden erinomaisen näkökulman kouluttaa nykyisiä ja tulevia projektipäälliköitä.

Seppo Pääjärvi

Why and how can human-related measures support software development processes?

Orit Hazzan and Irit Hadar, Journal of Systems and Software, volume 81, pages 1248-1252, 2008

Johdanto

Inhimillisiä seikkoja otetaan nykyään aikaisempaa enemmän huomioon ohjelmistokehityksessä. Näkyvin merkki siitä ovat ketterät ohjelmistokehitysmallit, joissa asiakas ja ihmisten välinen kommunikaatio ovat keskeisessä roolissa. Hazzan ja Hadar käsittelevät artikkelissaan ohjelmistokehityksessä apuna käytettäviä arviointi- ja mittausmenetelmiä. Heidän mielestään inhimilliset seikat tulisi ottaa nykyistä enemmän huomioon myös ohjelmistoprosessin laatua ja ominaisuuksia mitattaessa.

Tulokset

Hazzan ja Hadar pohtivat artikkelissaan ohjelmistokehitykseen liittyviä inhimillisiä seikkoja. He toteavat, että suurin osa ohjelmistoprojekteissa ilmenevistä ongelmista johtuvat juuri niistä. He arvioivat yleisimpiä ohjelmistoprosessin arvioimiseen käytettäviä teknisesti suuntautuvia menetelmiä sekä pohtivat niiden puutteita inhimillisten näkökulmien kannalta. Nykyiset menetelmät keskittyvät heidän mielestään liiaksi teknisiin ja johtamistaidollisiin näkökulmiin jättäen inhimilliset näkökulmat taka-alalle.

Kirjoittajat kertovat miten ohjelmistoprojekteissa esiin tulevia, inhimillisiin seikkoihin liittyviä ongelmia voitaisiin ratkaista. Ongelmiin paneutuminen vaatii jonkinlaisten inhimillisten ominaisuuksien arviointimenetelmien kehittämistä. He ehdottavatkin, että keskityttäisiin sosiaalisiin ja kognitiivisiin näkökulmiin yksilö- ja tiimitasolla.

Kirjoittajat jakaisivat kerättävän tiedon neljään erilaiseen ryhmään (käytännössä taulukon soluihin). Kaksi ryhmistä, kognitiivinen yksilö ja sosiaalinen yksilö keskittyvät yksilön ominaisuuksiin. Kaksi muuta ryhmää, kognitiivinen tiimi ja sosiaalinen tiimi, taas keskittyvät lähinnä tiimin sisäisiin ominaisuuksiin. Kognitiiviset ominaisuudet kuvaavat lähinnä yksilön omaa ja tiimin sisäistä oppimista. Sosiaaliset ominaisuudet puolestaan kuvaavat yksilön ja tiimin välistä sekä tiimin sisäistä kommunikaatiota. Kaikki ominaisuudet ovat projekteihin osallistuvien yksilöiden ja tiimien itse arvioimia.

Johtopäätökset

Kirjoittajien mielestä tulisi olla arviointimenetelmä, joka ottaisi tekniset, johtamistaidolliset ja inhimilliset seikat tasapuolisesti huomioon. Ohjelmistoprosessin aineettomuus verrattuna perinteisiin tuotantoprosesseihin on heidän mielestään keskeisin syy sille, että ohjelmistoprojektien yleisimmät ongelmat ovat ihmisten toimintaan liittyviä. Heidän kuvaamansa menetelmät auttaisivat tekemään ohjelmistojen ominaisuuksista konkreettisempia ja niin ollen helpompia ymmärtää.

Tiina Taivalantti

The impact of software process standardization on software flexibility and project management performance: Control theory perspective

Julie Yu-Chih Liu, Victor J. Chen, Chien-Lung Chan and Ting Lie, Information and Software Technology, volume 50, issue 9-10, pages 889-896, 2008

Johdanto

Vuosikausia on oletettu ohjelmistokehityksen prosessien standardoinnin parantavan ohjelmistoprojektien tehokkuutta. Oletukset perustuvat ajatukseen, että suunnitellut, sekä hyväksi koetut standardoidut kehitysmallit tehostavat ohjelmistoprojektia.

Jatkuva ohjelmistojen tarve lisää myös ohjelmistokehityksen tarvetta, minkä oletetaan lisäävän ohjelmistojen laatuongelmia. Tämä huolenaihe esiintyy tutkimuksissa, joissa on noussut esiin ohjelmistoprojektin tehokkuus pääasiallisena projektin johdon huolenaiheena.

Ohjelmiston laatua voidaan arvioida ohjelmiston joustavuuden perusteella. Joustavuus tarkoittaa helpottunutta ylläpitoa, helppoa ohjelman muokkaamista, ominaisuuksien lisäämistä, virheiden hallintaa ja helpompaa ohjelmiston päivitystä. Ohjelman joustavuus on siten yksi kriteeri kun arvioidaan ohjelmistoprojektin lopputuloksia.

Tutkimustuloksia siitä, mitä elementtejä ohjelmistoprojektin tulisi sisältää, jotta projektia voidaan pitää tehokkaana, ja standardoitua työtapaa hyödyllisenä, on melko vähän. Tutkimuksen tarve on olemassa, koska monet organisaatiot ovat, taistelllessaan budjetti- ja aikatauluylilyksiä vastaan, keskittäneet enemmän huomiota ohjelmistoprosessien parantamiseen (SPI – software process improvement).

Tulokset

Tutkimuksen tarkoituksena oli kartoittaa, mitkä prosessimallit ovat avainasemassa ohjelmistotuotantoprosessin standardoinnissa. Tutkimus suoritettiin lähettämällä kysely ohjelmistotuotannon ammattilaisille, jotka valittiin satunnaisesti niistä IEEE Computer Society -järjestön jäsenistä jotka tilaavat IEEE Software -lehteä.

Tutkimuksessa esitettiin kolme hypoteesia:

1. Prosessien standardoinnin ja projektin tehokkuuden välillä vallitsee vuorovaikutussuhde.
2. Prosessien standardoinnin ja ohjelmiston joustavuuden välillä vallitsee vuorovaikutussuhde.
3. Prosessien joustavuuden ja projektin tehokkuuden välillä vallitsee vuorovaikutussuhde.

Kyselyn tulokset osoittavat suoraan, että yhteys ohjelmistoprosessien standardoimisen ja projektin tehokkuuden välillä on merkittävä. Lisäksi kyselyn tulokset osoittivat että prosessien standardoinnilla on merkittävä yhteys ohjelmiston joustavuuteen. Ohjelmiston joustavuus vaikuttaa myös positiivisesti projektin tehokkuuteen.

Jotta esitetyt hypoteesit pitävät paikkaansa, täytyy prosessien standardoinnin edellyttää ohjelmiston joustavuutta sekä prosessien standardoinnin tulee edellyttää projektin tehokkuutta. Tämän lisäksi kun prosessien standardointia sekä ohjelmiston

joustavuutta tarkastellaan yhdessä projektin suorituskyvyn edellytyksinä, tulisi standardoinnin kerrannaisvaikutuksen olla pienempi kuin jos standardointi mallinnettaisiin yksinään.

Tuloksia tarkemmin analysoimalla saatiin tulokseksi, että kaikki kolme hypoteesia näyttävät pitävän paikkansa.

Johtopäätökset

Ohjelmisto prosessien standardointi johtaa projektin tehokkuuden parantamiseen ja ohjelmiston laatuun joustavuuden ollessa laatutekijä. Lisäksi ohjelmiston joustavuus juontuu prosessien standardoinnin ja projektin tehokkuuden suhteesta.

Tutkimuksen tulokset tulisi ottaa huomioon organisaatioiden prosesseja suunniteltaessa. Organisaatioiden tulisi ottaa käyttöön organisaatiotasoinen prosessimalli ohjelmistokehitykseen, tai ainakin implementoida prosessikehys käyttäen muutamia valittuja standardeja. Yhtenäinen prosessikehys parantaisi todennäköisesti kehitysprojektien tehokkuutta ja suorituskykyä läpi organisaation, sekä parantaisi henkilöstön tietotaitoa oppimisprosessin kautta, henkilöstön työskennellessä heille tuttuun prosessimallien mukaan. Ohjelmistojen laatu joustavuuden suhteen lisääntyy oletettavasti standardoitujen prosessien käyttöönoton myötä.

Timi Antere

Foundations of program management: A bibliometric view

Karlos Artto, Miia Martinsuo, Hans Georg Gemünden and Jarkko Murtoaro, International Journal of Project Management, volume 27, Issue 1, pages 1-18, 2009

Johdanto

Katsaus lähtee selvittämään kysymystä siitä, onko ohjelmistoprojektin johtaminen todella suoraan verrattavissa projektinjohtamiseen, vai onko ohjelmistoprojektin johtamisessa joitain sellaisia erityispiirteitä, joita muunlaisten projektien johtamisessa ei kenties esiinny. Työvälineenä katsauksessa on kirjallisuuskatsaus. Katsaus käsittelee 517 artikkelia ohjelmistoprojektin johtamisesta ja 1164 artikkelia projektinjohtamisesta. Kaikki artikkelit on julkaistu viimeisen 21 vuoden aikana johtavissa tieteellisissä julkaisuissa. Artikkelit paljastavat samankaltaisuuksia ja poikkeavuuksia teoreettisessa pohjassa lähteitä tarkasteltaessa, sekä teemoissa avainsanoja tarkasteltaessa.

Katsaus listaa yksitoista eriyttävää piirrettä ohjelmistoprojektien johtamisesta ja projektien johtamisesta.

Projektilla tässä tekstissä viitataan johonkin sellaiseen projektiin, joka ei ole ohjelmistoprojekti. Tällaisia voivat olla vaikkapa tuotantoprojektit, joissa suunnitellaan ja valmistetaan jokin konkreettinen fyysinen tuote.

Tulokset

Tutkimus toteutettiin kirjallisuuskatsauksella vertaamalla tiettyjä asioita julkaistuista artikkeleista sekä ohjelmistoprojektien, että muiden projektien johtamisesta. Katsauksessa painotettiin kysymyksiä siitä, minkälaiselle pohjalle ohjelmistoprojektien johtaminen artikkeleiden mukaan rakentuu, miten nämä pohjat eroavat ohjelmistoprojektien johtamisen ja muiden projektien johtamisen välillä, sekä miten ohjelmistoprojektien johtamisen pohjat ovat kehittyneet ajan kuluessa.

Tutkimuksessa tarkasteltiin myös sitä, minkälaisia teemoja ohjelmistoprojektin johtamiseen liittyy avainsanojen perusteella, miten ne eroavat muiden projektien johtamisesta kirjoitettujen artikkeleiden avainsanoihin ja miten nämä teemat ovat muuttuneet ajan kuluessa.

Tutkimusaineiston lähteeksi valittiin kaikkiaan 23 johtavaa tieteellistä julkaisua sellaisilta aloilta, joihin todennäköisesti liittyy projektien johtaminen. Artikkelit haettiin hakusanoilla 'program' ja 'programme' rajattaessa haku vain ohjelmistoprojekteihin, sekä 'project' haettaessa yleensä projektin johtamista käsitteleviä artikkeleita. Hakuun sisällytettiin artikkelit aikaväliltä 1986 – 2006, mutta kirjoitusajankohdan vuoksi vuoden 2006 artikkeleista vain alkuvuonna julkaistut ehtivät mukaan.

Tutkimuksessa havaittiin, että projektit viittaavat vahvimmin tuotteen kehittämiseen, kun taas ohjelmistoprojektit viittaavat laaja-alaisesti eri ohjelmistoprojektin johtamisen osa-alueisiin, kuten valmistus, laadunvarmistus ja organisaatorakenne.

Ohjelmistoprojektien johtamisen teemat vaikuttivat vaihtelevat ajan kuluessa, kun taas projektijohtamisen teemat pysyivät enemmän samankaltaisina. Molemmissa tutkituissa johtamisen aloissa teemat kuitenkin osoittivat huomattavaa samankaltaisuutta, joskin hieman erilaisilla painotuksilla. Onnistuminen oli teema joka löytyi vain projektinjohtamista käsittelevistä artikkeleista.

Yhteenveto

Ohjelmistoprojektien johtamista ja projektienjohtamista käsittelevissä artikkeleissa oli paljon yhteneväisyyksiä ja teemojen samankaltaisuuksia, joka oli mielestäni odotettavissakin. Tutkimuksessa havaittiin kuitenkin tiettyjä poikkeavuuksia tai erityispiirteitä kahden kohteen välillä. Oletettavasti merkittävin ero löytyykin kirjoitusten näkökannoista, mutta se on asia, jota on vaikea tuoda esiin tilastollisella analyysillä.

Antti Sand

A feasibility evaluation on the outsourcing of quality testing and inspection

Minsoo Choi, Michael Brand and Jinu Kim, International Journal of Project Management, volume 27, pages 89-95, 2009

Johdanto

Artikkeli kertoo tutkimuksesta, jossa arvioitiin mahdollisuutta ulkoistaa testaus ja tarkastusaktiviteetit rakennusosalalla. Tutkimus perustuu Koreassa vuonna 2006 tehtyyn mielipidekyselyyn, johon osallistui 22 laadunhallinnan ammattilaista, ja sen pohjalta tehtyyn arviointiin AHP (Analytic Hierarchy Process) -menetelmällä.

Tulokset

Aluksi määriteltiin kriteerit, joiden perusteella ulkoistamista arvioitiin. Kriteereitä löydettiin viisi:

- 1) Mahdollisuus suorittaa testaus/tarkastuksia systemaattisesti
- 2) Testaajan/tarkastajan ammattitaidon lisääntyminen
- 3) Objektiivisuuden lisääntyminen
- 4) Vastuunkannon turvaaminen
- 5) Testaustekniikoiden statuksen parantuminen

Vastausten tuloksia käsiteltiin näiden kriteereiden valossa AHP -menetelmällä. Tuloksista kävi ilmi, että suurin osa vastaajista oli sitä mieltä, että testauksen/tarkastuksen ulkoistaminen on tarpeellista ja suositeltavaa. Ulkoistaminen lisää testauksen/tarkastuksen objektiivisuutta ja testaajien/tarkastajien ammattitaitoa ja arvostusta. Ulkoistamisen negatiivisina puolina nähtiin, että se voi vähentää testaushenkilöstön vastuutuntoa ja -ottoa. Vastauksista kävi myös ilmi, että yleisesti ottaen laadunhallintaa ei tule ulkoistaa, paitsi pienillä työmailla. Tärkeää on erottaa laadunhallinnasta omaksi alueekseen ”laadun testaus ja tarkastus” -toiminta, jonka ulkoistamisesta ja suorittavan yrityksen valinnasta päättää omistajataho, ei urakoitsija.

Johtopäätökset

Vaikka artikkelissa käsiteltiinkin testauksen ulkoistamista rakennusalan näkökulmasta, voi saatuja tuloksia monelta osin soveltaa myös IT -alaan. IT -alallakin riippumaton, ulkoistettu testaus parantaa objektiivisuutta, on ammattitaitoista, eikä sido yrityksen omia resursseja.

Piia Sajasalo

Towards a conceptualisation of PMO's as agents and subjects of change and renewal

Sergio Pellegrinelli and Luciano Garagna, International Journal of Project Management, volume 27, pages 649 – 656, 2009

Johdanto

Artikkeli kertoo PMO:den (Project and Programme Management Offices eli projektinhallinta toimistojen/osastojen) yleisestä määrittelystä ja siitä millaisena kirjallisuus ne esittää ja kuinka tätä näkemystä pitäisi muuttaa, että se vastaisi paremmin todellisuutta.

Tulokset

Vuonna 2007 lokakuussa kutsuttiin koolle foorumi, jonne pyydettiin seitsemän edustajaa. Edustajat olivat toimineet kauan projektinhallintatehtävissä suuryrityksissä. Edustajia pyydettiin vertailemaan kokemuksiaan PMO:sta ja niiden tarpeellisuudesta sekä luontees-ta. Paikalla oli myös henkilö, joka kirjasi kommentit ylös myöhempää analysointia varten.

Edustajat olivat yhtä mieltä siitä, että PMO:t olivat keskenään varsin erilaisia ja niiden tehtävät sekä toimenkuvat erosivat paljon riippuen yrityksestä ja yrityksen toimialasta. Yleisesti todettiin, että PMO:den elinkaari jäi usein vain 3-4 vuoden ikäiseksi, mutta se saatettiin myös perustaa uudelleen myöhemmin.

Tämä tosiasia eroaa suuresti kirjallisuudessa yleisesti annettavaan kuvaan, että jokaisessa yrityksessä pitäisi olla jatkuvasti toimiva, projektin johtoa valvova elin, jonka tehtävinä ovat mm. prosessien standardointi, budjetointi, resursointi ja riskien hallinta sekä yleisesti projektipäälliköiden tukeminen.

Tällainen näkökulma johtaa kuitenkin ongelmiin, sillä PMO:n siirrettyä tietotaitonsa sekä uudet työskentelymallit alas käytännön tasolle, sen olemassaololle ei ole enää oikeutusta ja näin ollen PMO lakkautetaan.

Johtopäätökset

Forumista kerättyjen löytöjen pohjalta tehtiin ehdotus PMO:n määritelmän uudistamisesta. Sen sijaan, että PMO nähdään kiinteänä ja muuttumattomana osana yrityksen organisaatiota, se pitäisikin nähdä uudistuvana ja toimenkuvaansa kehittävänä elimenä. Kun tarve - jota vastaamaan PMO alun perin luotiin - poistuu, se voi muuttaa toimintastrategiaansa ja kehittää itseään vastaamaan uusiin haasteisiin. Näin se saa jatkuvasti uusia kohteita, joilla tuoda lisäarvoa yritykselle.

Jaana Partanen

How do personality, team processes and task characteristics relate to job satisfaction and software quality?

Silvia T. Acuña, Marta Gómez and Natalia Juristo, Information and Software Technology, volume 51, pages 627-639, 2009

Johdanto

Artikkeli analysoi ihmisten persoonien, ryhmän prosessien, tehtävän luonteen, ohjelmistotuotteen laadun ja jäsenten tyytyväisyyden välisiä suhteita ohjelmistonkehitysryhmissä. Tutkimuksen tarkoituksena on selvittää mitkä tekijät edistävät kehitysryhmän tuotteliaisuutta, jotta projektipäälliköt voivat hyödyntää tätä tietoa ryhmien muodostamisessa.

Tulokset

Tutkimus toteutettiin vuonna 2008 ja siihen osallistui 105 toisen vuoden tietojenkäsittelytieteiden opiskelijaa, jotka jaettiin satunnaisesti kolmen hengen ohjelmistonkehitysryhmiin. Ryhmien tehtävänä oli toteuttaa ohjelmisto ketterän kehitysmallin mukaisesti käyttäen Extreme Programming-menetelmää. Toteutetun ohjelmiston laatua arvioitiin standardeilla ohjelmistotuotannon kriteereillä ja osallistujien persoonallisuutta, yhtenäisyyttä, ryhmien konflikteja, tehtävien autonomiaa sekä ryhmän tyytyväisyyttä arvioitiin erilaisilla Likert-asteikkoa hyödyntävillä valmiilla kyselylomakkeilla.

Tutkimuksen tuloksena saatiin käytännönläheisiä vinkkejä projektipäälliköille ryhmien muodostamista varten: Henkilöstön ulospäinsuuntautuneisuuden taso tulisi mitata kyselylomakkeilla, ja järjestää ryhmät siten, että ryhmissä tämä taso olisi keskitasoa. Valittavien tehtävien tulisi olla toisistaan riippuvaisia ja työtapojen tulisi tukea tätä. Ohjelmiston kehitysprosessin aikana tulisi myös seurata mahdollisia ryhmän sisäisiä konflikteja sekä ryhmän jäsenten tyytyväisyyttä ja reagoida tarpeen mukaan näihin.

Yhteenveto

Tutkimuksen mukaan työhönsä kaikkein tyytyväisimmät ryhmät ovat sellaisia, joissa on tunnollisia ja myöntyväisiä jäseniä. Näyttää myös siltä että ekstroverttiydellä on suuri merkitys ohjelmiston laatuun käytettäessä ketteriä kehitysmalleja.

Tutkimus on vielä hyvin suppea eikä siksi suoraan hyödynnettävissä oikean elämän tilanteisiin, mutta se toimii hyvänä lähtökohtana tulevaisuuden tutkimuksille.

Jonna Paananen