Quality issues in agile software development

Elena Soloveva

University of Tampere
School of Information Sciences
Degree Programme in Software Development
Elena Soloveva: Quality issues in agile
software development
M. Sc. Thesis Proposal, 18 pages, October
2017

1st Supervisor
Name and Surname: Eleni Berki
e-mail: cselbe@sis.uta.fi

Abstract

Agile manifesto transforms developers' mindset, enabling them to make rapid deliveries. Lack of conceptual clarity in the agile theory leads to multi-faceted agile nature, with teams interpreting and implementing agile variously. The team's decisions present paramount importance in the agile setting, with poor decision making quality being a significant factor in causing projects' failures.

The goal of this study is to understand the challenges agile manifesto presents for the developers and identify quality issues agile teams could observe in practice when using agile. Implications of agile manifesto will be outlined based on its critical analysis. Furthermore, agile team's decision making, as a crucial agile challenge, will be investigated. The quality-centric survey will be designed and one of the agile teams in Finland is going to participate in this case study. The collected data will be analysed revealing the obstacles in agile manifesto application observed by this team, in terms of quality issues.

Understanding agile manifesto challenges and quality issues in agile usage is vital to increase success of the agile usage in teams. The findings will also highlight the gap in the agile theory that the team / research need to be aware of and find optimal solutions.

Key words and terms: agile manifesto challenges, decision making, agile quality issues.

Contents

Abstract....................i
1. Introduction....................1
2. Literature Review. Decision Making as Agile Usage Challenge....................3
2.1. Agile manifesto challenges....................3
2.1.1. “Individuals and Interactions over processes and tools” [Agile Alliance, 2001]...3
2.1.2. “Customer Collaboration over contract negotiation” [Agile Alliance, 2001]........4
2.1.3. “Responding to Change over following a plan” [Agile Alliance, 2001]................5
2.1.4. “Working software over comprehensive documentation” [Agile Alliance, 2001] 6
2.2. Overview of decisions in Agile Software Development....................7
2.2.1. Descriptive decision making....................7
2.2.2. Decision making in Agile context....................8
3. Design and methodology....................9
3.1. Motivation and purpose of the case study....................9
3.2. Data collection methodology....................10
References....................11

1. Introduction

Agile methodology provides set of practices for software development. However, there is a lack of rigorous guidelines describing how to implement agile in the projects to achieve fast deliveries [Conboy, 2009]. Moreover, agile manifesto sets challenges for the software development teams, thus moving from agile theory to practice is not straightforward. Extent of achieved agility should be assessed within the context in which agile method is applied [Alleman, 2002]. Each project is unique and in this regard, it adopts agile in its own way, depending on the team's understanding of agile, consideration for the needed set of practices and approaches how to apply them.

Software product is not a result of single individual work, but the outcome of collaborative team contributions. Traditional methods emphasized focus on achievements of each individual, creating competitive atmosphere, rather than moving to the project goal as a collaborative team [Hackman, 1987]. In contrary, in agile software development the team takes the central role and is given the ownership for the software development decisions [Agile Alliance, 2001].

Research Question 1: What are the challenges agile sets for the agile teams?

1.1 What kind of implications agile manifesto statements present for the team?

1.2 How do agile values influence decision making in agile team?

Research Question 2: What are the quality issues agile teams may face when implementing agile? (Case Study)

2.1 What are the quality issues in the team's decision making?

2.2 What are other quality issues and their correlation with decision making issues?

The chosen process of software development and selected methods highly contribute to the quality of the resulted product [Carleton et al., 1994; Acton et. al., 2014]. However, people are still most influencing software engineering factor [Cockburn, 2001]. With emergence of agile software methodologies, the original focus in software engineering has moved from process to people determining the project success [Broza, 2012]. Since focus in defining methodologies was on process rather than its use by practitioners, there is an evident gap as this approach neglects interpretation by people and human factor [AlQaisi, Grai and Steves, 2017].

The research of quality issues in agile software development is limited, with few studies looking into concerns of quality in agile methodology and highlighting difficulty to apply metrics to soft or uncontrollable factors prevailing in agile, for instance measurements of knowledge creation and transfer [Berki, Siakas and Georgiadou, 2007]. In this thesis work, agile quality is evaluated from the aspects of how teams are successful in interpretation of agile manifesto core values, in their approaches to satisfy its ambitions, such as achieving team collaboration and team decisions quality on numerous factors arising throughout the software development process.

The project success strongly relies on the ability of the team to identify quality issues and implement corrective actions before they turn into project failures to support dynamic software evolution demands [Laplante and Neill, 2006]. Understanding the obstacles that the agile team may face when using agile methodology is instrumental for practitioners to reveal and address corresponding issues.

Considering importance of the context in agility evaluations, the empirical study is instrumental to identify quality issues in the team’s decision making and how outcomes of such decisions turn into quality issues in user stories and tasks, architecture design, delivery infrastructure support, code inspection and tests. The study is centered on technical decisions quality owing to great importance on solutions quality and system scalability to provide fast deliveries on top of already implemented. The issues will showcase the obstacles teams face when applying agile manifestoo core values and principles in practice. Quantitative and qualitative methods will be used for the collected data analysis.

A literature review is conducted with critical analysis of agile manifesto and focus on decision making as a pivotal challenge in agile setting, with decisions eventually determining the project success.

This research is focused on collaboration and decision making, primarily in the technical meetings. Case study is limited to analysis of data collected based on the survey taken by a single team. While it does not reveal all potential issues teams can face when implementing agile, the issues identified in agile manifesto implementation raise awareness of the quality risks agile teams take. Moreover, when the issues are identified, it is vital for the teams to be aware of how to address them. This is out of the current research scope.

2. Literature Review. Decision Making as Agile Usage Challenge

This chapter examines existing research on agile challenges and decision making quality in agile software development. It starts with analysis of agile values statements available in the literature, and then flows into comprehension of decision making models. Applying of agile values into practice greatly relies on abilities of agile teams to make quality decisions, which requires broad range of skills, such as people-oriented (collaboration / communication), brainstorming, planning, continuous integration skills and so on. By looking into decision making process, the impact on decision quality can be clarified [Dewan and Hansen, 1994]. This, in turn, is crucial, as poor decisions possess a software quality risk, including incomplete features, unexpected defects, unreliable estimates [Drury, Conboy and Actonb, 2017].

Based on the literature review, the survey will be designed in order to get insight into how one of the teams in Finland implements agility values, considering the decision making and quality issues the team members think they have in the project compared to the agile theory.

2.1. Agile manifesto challenges

The teams' decisions present paramount importance in the agile setting. Agile manifesto intends to transform the team mindset in order they could incorporate software changes fast. The agile values statements interpretation and application into work is under teams considerations. The manifesto indicates what are the highest priorities when need to make a trade-off to support a quick change. For example, individuals and interactions are over processes and tools [Agile Alliance, 2001]. Since it is not a concrete solution but rather suggested way of thinking, there is a freedom in understanding the proportions of each compound when it comes to practice. For example, to which extent planning should be reduced, to which extent processes are needed, etc. Hence, the four key agile values present challenges in the decision making [Drury, Conboy and Actonb, 2017]. To aid developers in grasping the values captured in Agile manifesto, the Alliance clarified them into 12 principles, which are discussed below [Agile Alliance, 2001].

2.1.1. "Individuals and Interactions over processes and tools" [Agile Alliance, 2001]

The following agile principles relate to a teamwork and it can be noticed, that there is a strong highlight of the *self-organizing and self-improving* as a team essential properties:

“*The best architectures, requirements, and designs emerge from self-organizing teams.*
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Build projects around motivated individuals.
Give them the environment and support they need, and trust them to get the job done.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”

The statements convey that team work is paramount in agile settings and the team should be able to:

1. *Self-organise.* The team is gradually learning and gaining management and technical skills to take decisions, which previously were outside of traditional developers skills. It can create situation where the team does not yet have skills or knowledge for effective decision taking. It is also unclear to which extent teams are looking for the information and skills development when taking decisions. [Drury, Conboy and Actonb, 2017]
2. *Effectively collaborate.* Knowledge sharing on the system (UI development, architecture, testing, infrastructure, etc.) occurs through high *team collaboration*. There is no specified roles, the knowledge is not narrowly concentrated, but instead shared between everyone in the team and roles are merged. At the same time, expertise in different areas feeds finding optimal decision [Hackman, 1987; Seers et al., 1995].
3. *Improve work.* Daily collaboration, and retrospective meeting in the end of the iteration serve the purpose of reflection on the team work and identification of improvements. However, people relate causes to what they see rather then what is operating behind the hood [Hackman, 1987]. The issues in decision making may be hard to reveal and thus improve. The teams also tend to take repetitive decisions, based on the previous experience (). Thus, the team may stick to the previous choices instead of analyzing the situation. This also means the team may be on the course of certain way of thinking all along the process, not being able to realize weak points and thus improve.

2.1.2. “Customer Collaboration over contract negotiation” [Agile Alliance, 2001]

The following agile manifesto principles relate to interactions with customer [Agile Alliance, 2001]:

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Business people and developers must work together daily throughout the project.”

Delivery of valuable software assumes high customer motivation in the development of the product [Chow and Cao, 2008], as well as being knowledgeable about the domain [Qasaimeh et al., 2008]. In addition, according to Qasaimeh study, the approach to customer collaboration varies between Agile methods, such as SCRUM, XP, FDD and Crystal. Depending on the degree of customer involvement and competence in the domain area, the projects may choose the most suitable Agile methodology practice [Kasem and Razali, 2017].

In case customer is active and have some vision on the product, then it is easier to define what the next expected valuable solutions are and receive constructive feedback. Not only the customer, but the team can suggest possible deliverables, and align the content and schedule with the customer. This is the concept of collaboration, and it is encouraged to be continuous to provide correct set of functionalities. [Kasem and Razali, 2017]

Insufficient or not effective customer involvement can lead to a breakdown in the collaborative atmosphere [Coffin, 2016]. At the extreme side, there are projects where the team has to take complete ownership to identify the deliverables by themselves. Then the team is highly responsible to determine what is valuable and prioritize work accordingly. It showcases that ability for the team to identify what user stories are the most valuable is vital and for some projects is crucial.

2.1.3. “Responding to Change over following a plan” [Agile Alliance, 2001]

The following agile principle regards the changes handling, specifically addressing the case when they interrupt planned activities: *“Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.”* [Agile Alliance, 2001]

Software development methods can vary from being adaptive to, opposite, predictive. The approach to planning in the project is one of the key characteristics influenced by this factor. Agile, being adaptive, gives flexibility in how to achieve goals, while predictive software development methods focus on detailed planning.

Agile teams may not know which tasks they will do next week, more prevalent is instead having a clear closest milestone to achieve [Cockburn, 2001].

Agile manifesto suggests “Responding to Change over following a plan”. This puts teams’ mindset towards making everything to facilitate fast adaptivity to a change, which means less strict planning. The extent of planning reduction should be defined keeping in mind other agile development principles, like continuous improvement, good design and architecture.

Since multiple variables are involved in this equation, teams decisions on what minimal planning varies in the projects.

2.1.4. “Working software over comprehensive documentation” [Agile Alliance, 2001]

The following agile principles relate to the documentation handling and importance of frequent deliveries:

“Working software is the primary measure of progress.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Simplicity--the art of maximizing the amount of work not done--is essential.
Continuous attention to technical excellence and good design enhances agility.”

The goal of an agile team is to incorporate agility in the software development process, releasing the product often. The agile principles advocate attention to the “technical excellence” and “good design”, which in its turn imply higher cost for solution design and implementation. Hence, this creates equation for the team to find for each technical solution a balance between agility and technical excellence.

Focus on providing higher value implies concentrating only on use case that brings value right now. This means also while designing the solution, the team may decide that consideration of affected / generated by the solution boundary use cases is secondary. As a result, the team may not reveal impacts on the architecture and its stability in that case. Such technical decisions of the team which favor shortcuts in the solution design phase may incur technical debt. The term technical debt was initially introduced by Ward Cunningham, which refers to omitted quality in the solutions [Cunningham, 1992]. Buschmann also identified that not only this can happen unintentionally

[Buschmann, 2011], but some teams use technical debt to accelerate the development pace. In the conducted by Buschmann case study, this led to major performance issues that turned later on to business consequences.

2.2. Overview of decisions in Agile Software Development

Agile positions teamwork as a collaborative work, replacing the concept of individual work focus which is used in traditional development [Vinekar et al., 2006]. Consequently, agile development compared to traditional introduces:

- multidisciplinary vs specialized skillset
- high collaboration vs individual work
- pluralist decision making vs manager as accountable for decisions.

Agile software development teams become at the front of taking critical decisions that eventually result in the project success or failure [Conboy, 2009]. At the same time, the project managers are less involved as decision makers [Alleman, 2002], empowering agile teams to take decisions. In contrary to decline of project manager's influence, customers still greatly impact the product development decisions in agile teams [Beck, 2000].

Agile team engages in numerous decision taking during software development cycle [Drury et al, 2011].

2.2.1. Descriptive decision making

Descriptive decision making (DDM) suggests decision making should be studied considering the context. This enables to reveal aspects influencing decision makers' behavior in addition to the ones already specified research, such as perception and amount of uncertainty, extent of information use, tendency to re-use old approaches / solutions, complexity of the problem and internal conflicts.

The researchers concluded that in the cases when decision making is concerned with multiple choices involving numerous attributes, decision model can be characterized as compensatory or non-compensatory [Payne, 1976; Tversky, 1972]. A compensatory decision making implies comparison of the variables values of each option in order to identify optimal alternative. In this principle of decision making high value of one variable can compensate lower value of another.

In contrary, non-compensatory decision making implies shortcuts of compensatory strategy by simplifying variables evaluation. In this tactic selection of the best alternative is achieved through elimination of choices by variables threshold values. This makes the decision faster providing less efforts are needed to evaluate variables. The drawback is typically a resulting decrease in decision quality as not all available information is examined.

Decision makers tend to choose a decision making strategy based on many aspects, involving cognitive abilities, experience and information accessibility. The need to use more accurate strategies arises when the mistake in the decision has significant impact. In this situation inaccurate decision is not justified at the expense of saving time [Payne et al., 1988].

2.2.2. Decision making in Agile context

In the agile settings, there is an accuracy-effort trade-off between taking the time to generate and evaluate alternatives and time saving in favour of fast outcome. Agile teams’ decision making process exhibits use of minimum information, reduced analysis and comparing of bad and good aspects of alternatives [McAvoy and Butler, 2009]. This showcases the cutting-off in the considered variables in sake of easier to derive and most important variables. It means agile teams employ non-compensatory decision strategy [Drury et al, 2017].

There is still little known about the obstacles ASD team encounter in the decision making.

Lipshitz and other researchers looked into characteristics of decision making in practice, and admitted that teams do not necessarily generate all possible options. Moreover, further comparing them may lack evaluation criteria and system [Lipshitz et al., 2001].

Experience

Research revealed that experience is a driving force in ASD project management decisions [Drury et al., 2011].

Low team members contribution

It was discovered that poor decision making can be a result of low contribution of team members in the discussions, in case they are not sharing opinions which are opposite to other team members [McAvoy and Butler, 2009].

Limited information

Also, actual information is not being inquired and utilized at full in favor of saving time. Agile context creates situations for suboptimal decision processes such as groupthink, groupshift, or a liability to defer awkward decisions until the future [Briggs and Reinig, 2010; Coyle et al., 2011].

3. Design and methodology

3.1. Motivation and purpose of the case study

Considerable amount of existing research relates to positive factors of agile methodologies [Highsmith and Cockburn, 2001], with some covering challenges when migrating to agile [Boehm, 2002], yet there is little focus on the issues teams experience in the practice. Since agile process is empirical, as opposite to defined processes, there is no universal solutions but instead customized approach is required for each project case.

Collaboration is core in agile settings [Agile Alliance, 2001]. This assumes not only communication between team members, but also ability to work together to achieve project goal [Highsmith and Cockburn, 2001]. The way how teams make decisions indicates the extent of collaboration [Highsmith, 2001].

DDM research summarised several principles as factors affecting decision making, namely uncertainty, limited information, behaviors which are adapted, complexity of the problem or internal conflict in decision making, and contextual differences. Drury et al. investigated decision process in several agile teams, with the case studies which revealed issues like team members relying on others for decisions and team members not taking ownership, and collaborative decision making restraining experts opinions. [Drury et al., 2012].

While Drury and O’Dwyer studies are focused on agile planning and daily meetings [Drury and O`Dwyer, 2013], the goal of this research was investigation in technical meetings. As it was previously noted, some agile teams tend to have solutions design in smaller groups and out of agile planning meetings. The specific interest is the solutions that impact architecture or API design, systems scalability, as those impact agility of adding further changes to the system. The quality of technical meetings decisions impacts agility pace and quality of the resulted product.

The current study extends agile research by exploring agile team challenges with focus on people opinions which in practice define customised approach for applying agile in the project. The idea is to understand both perspectives, how they themselves identify issues individually, and at team level. It may also clarify why even at the time when team members have bright ideas, not everything is realized in practice.

Knowledge of the pitfalls experienced by agile teams is beneficial for the teams success. Moreover, revealing of such issues by teams on its own is hard as some issues relate to and require change of the team mindset. This increases value of awareness of possible obstacles that they may face.

3.2. Data collection methodology

A survey was designed to address research questions. The questionnaire offered multiple and single choice, rating and open ended questions. It started with explanation of its purpose and security of data collected, highlighting also that the participation is voluntary and the person could stop the survey at any time if does not want to proceed by any reason. The survey was divided into three main sections. Overall, there were 22 questions in the survey.

The first survey's section is general, and inquires team members experience in software development, agile projects, implementing microservices architecture and maintaining infrastructure supporting deliveries.

The second survey's section aims at getting insight into a process of decision making in the team. This includes questions about size of the group taking decisions, information sharing, communication issues, satisfaction with the taken decision and decisions intelligence. The technical solutions design clarification is the desired outcome of technical decisions. The decisions quality in that case reflects technical solutions accuracy, and if there are shortcuts in the design, it would indicate increased likelihood of solutions removal. This part of the survey showcases also the challenges posed by inaccuracy of Agile manifesto in its formulation. Overall, the designed questions reveal the teams approach to implement the agile values in practice.

The concluding survey's section examines team readiness for fast deliveries. The questions cover user stories clarity and breaking into tasks, ability to solve infrastructure issues. Importantly, this gets insight in how team visions a need of imprvements by asking team members opinion on the need of changes in microservices architecture, testing and code review practices.

Data collected will be analysed using quantitative and qualitative methods.

The received data will be grouped and referred to the agile vaues statements to determine how each of the agile values has transformed into team mindeset in practice. Only those questions will be discussed, which answers assisted in finding correlation with agile mindset and decision making, as highly influenced by agile manifesto principles.

References

[Acton et. al., 2014] Acton, D., Kourie D., Watson B. Quality in software development: A pragmatic approach using metrics, *SACJ*, 52

[Agile Alliance, 2001] Manifesto for Agile Software Development. Retrieved 30th October 2017, from the Agile Alliance: http://agilemanifestoo.org/

[Alleman, 2002] Alleman, G. Agile Project Management Methods for IT Projects. In *The Story of Managing Projects: A Global, Cross-Disciplinary Collection of Perspectives* (Carayannis, E.G. and Kwak, Y.H, eds.), Greenwood Press, Berkeley, CA.

[AlQaisi, Grai and Steves, 2017] AlQaisi, R., Gray, E. and Steves, B. Software Systems Engineering: A Journey to Contemporary Agile and Beyond, Do People Matter? In: *Achieving Software Quality in Development and Use. SQM*, 1 (25). BCS, Southampton, UK.

[Ambler, 2013] S. Ambler. Scaling agile: the Software Development Context Framework, Available: *http://www.disciplinedagiledelivery.com/sdcf/.*

[Aydinli, Berki et al., 2016] Management anti-patterns in IT innovation projects, *Proceedings of the 20th International Academic Mindtrek Conference*, Tampere, Finland, 1-10.

[Barron, 1974] Behavioral decision theory: a topical bibliography for management scientists, *Interfaces* 5, 56-62.

[Beck, 2000] Beck, K. *Extreme Programming Explained: Embrace Change*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

[Bell et al., 1988] Decision Making: Descriptive, normative and prescriptive interactions in decision making, *Cambridge University Press*, Cambridge, 9-30.

[Berki et al., 2007] Berki, E., Isomäki, H. and Salminen, A. Quality and Trust Relationships in Software Development, *Proc SQM2007.*

[Berki, Siakas and Georgiadou, 2007] Berki, E., Siakas, K. and Georgiadou, E. (2007). Agile quality or depth of reasoning? Applicability vs. suitability with respect to

Stakeholders' needs, In: *Agile Software Development Quality Assurance*, Idea Publishing, 23-55.

[Boehm, 2002] Boehm, B. Get ready for agile methods, with care, *IEEE Software*, 35(1), 64-69.
[Boehm et. al., 2005] Boehm, B., Turner, R. Management Challenges to Implementing Agile Processes in Traditional Development Organizations, *IEEE Software*, 22(5), 30-39.

[Buschmann, 2011] Buschmann, F. To Pay or Not to Pay Technical Debt. *IEEE Software*, 28(6), 29–31.

[Chow and Cao, 2008] Chow, T. and Cao, D. B. A survey study of critical success factors in agile software projects", *Journal of Systems and Software*, 81, 961-971.

[Cockburn, 2001] Cockburn, A. *Agile Software Development*, Addison Wesley, Reading, MA.

[Conboy, 2009] Conboy, K. Agility from first principles: reconstructing the concept of agility in information systems development, *Information Systems Research*, 20(3), 329-354.

[Coffin, 2016] Coffin, R. A tale of two projects [agile projects], *Agile Conference*, 7-164.

[Cunningham, 1992] Cunningham, W. The WyCash Portfolio Management System. In: *Addendum to the Proceedings on Object-Oriented Programming Systems, Languages, and Applications*, OOPSLA, 29–30. ACM, New York.

[Dewan and Hansen, 1994] Rajiv M. Dewan, Stephen C. Hansen The role of user capability and incentives in group and individual decision support systems: An economics perspective, Decision incentives in group and individual decision support systems: An economics perspective, *Decision Support Systems*, 12(1), 25-40.

[Drury, Conboy and Actonb, 2017] Drury-Grogana, M. L., Conboy, K. and Actonb, T. Examining decision characteristics & challenges for agile software development, *Journal of Systems and Software*, 131, 248-265.

[Drury et al., 2011] Factors that Influence the Decision Making Process in Agile Project Teams Using Scrum Practices, *International Conference on Information Systems, Proceedings: The 6th International Research Workshop on IT Project Management.*

[Drury et al., 2012] Obstacles to Decision Making in Agile Software Development Teams, *The Journal of Systems and Software*, www.elsevier.com/locate/jss.

[Drury and O`Dwyer, 2013] An Investigation of the Decision-making Process in Agile Teams, *International Journal of Information Technology & Decision Making*, 12(6), 1097-1120.

[Duvall et al., 2007] Duvall, P.M., Matyas, S. and Glover, A. *Continuous integration: Improving Software Quality and Reducing Risk,* Pearson Education.

[Garvin, 1994] Garvin, D. What does product quality really mean. *Sloan Management Review*, 26(1), 25-34.

[Hackman, 1987] Hackman JR. The design of work teams. In: *Lorsch J Handbook of organizational behavior*. Englewood Cliffs, NJ: Prentice-Hall.

[Hare, 2017] Hare, C. *Designing for naturalistic decision-making,* PhD thesis, University of Nottingham.

[Carleton et al., 1994] Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., & Zubrow, D. (1994). *Benefits of CMM-based software process improvement: Initial results. Technical Report*, CMU/SEI-94-TR-13, August 1994.

[Highsmith and Cockburn, 2001] Highsmith, J. and Cockburn, A. Agile software development: The people factor. *IEEE Computer*, 34(9), 131-133.

[Kasem and Razali, 2017] Mashal Kasem Alqudah, Rozilawati Razali. Key Factors for Selecting an Agile Method: A Systematic Literature Review, *International Journal on Advanced Science, Engineering and Information Technology*, 7(1), 526-537.

[Laplante and Neill, 2006] Laplante, A.P. and C J. Neill (2006). *Anti-patterns: Identification, Refactoring and Management.* Auerbach Publications.

[Lipshitz et al., 2001] Taking Stock of Naturalistic Decision Making, *Journal of Behavioral Decision Making* 14, 331-352.

[McAvoy and Butler, 2009] The Role of Project Management in Ineffective Decision Making within Agile Software Development Projects, *European Journal of Information Systems* 18, 372-383.

[Misra et al., 2009] Misra, S. C., Kumar, V. and Kumar U. Identifying some important success factors in adopting agile software development practices, *Journal of Systems and Software*, vol. 82 (11), 1869-1890.

[Muhonen and Berki, 2011] An Open Process for Quality Assurance in Systems Development, *19th Intenational Conference on Software Quality Management, SQM 2011*.

[Payne, 1976] Payne, John, Task Complexity and Contingent Processing in Decision Making: An Information Search and Protocol Analysis, *Organizational Behavior and Human Performance*, 16, 366-387.

[Payne et al, 1993] Payne, John & Johnson, Eric & Bettman, James, *The Adaptive Decision Maker*, Cambridge University Press: Australia.

[Qasaimeh et al., 2008] Qasaimeh, M., Mehrfard H. and Hamou-Lhadj A. Comparing Agile Software Processes Based on the Software Development Project Requirements. *Computational, Intelligence for Modelling Control & Automation, International Conference*, 49-54.

[Simon, 1979] Rational Decision Making in Business Organisations, *The American Economic Review*, 69, 493-514.

[Spinellis, 2016] Diomidis Spinellis, Being a DevOps developer, *IEEE Software*, 33(3), 4-5.

[Vinekar et al., 2006] Vinekar, Vishnu, Craig W. Slinkman, and Sridhar Nerur. Can agile and traditional systems development approaches co-exist? An ambidextrous view. *Information Systems Management*, 23(3), 31-42.

[Woolley et al., 2008] Woolley AW, Gerbasi ME, Chabris CF, Kosslyn SM, Hackman JR. Bringing in the experts: How team composition and collaborative planning jointly shape analytic effectiveness. *Small Group Research*, 39, 352-371.