

Transitioning to Agile: A Framework for Pre-Adoption Analysis using Empirical Knowledge and Strategic Modeling

By

Hesam Chiniforooshan Esfahani

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Copyright © 2012 by Hesam Chiniforooshan Esfahani



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-97842-9

Our file Notre référence
ISBN: 978-0-494-97842-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada[!]

Transitioning to Agile: A Framework for Pre-Adoption Analysis using Empirical Knowledge and Strategic Modeling

Hesam Chiniforooshan Esfahani

Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto

2012

Abstract

Transitioning to the Agile style of software development has become an increasing phenomenon among software companies. The commonly perceived advantages of Agile, such as shortened time to market, improved efficiency, and reduced development waste are among key driving motivations of organizations to Agile. Each year a considerable number of empirical studies are being published, reporting on successful or unfavorable outcomes of enacting Agile in various organizations. Reusing this body of knowledge, and turning it into a concise and reachable source of information on Agile practices, can help many software organizations which are at the edge of transition to Agile, dealing with the uncertainties of such a decision.

One of the early steps of transitioning to Agile (or any other process model) is to confirm the adaptability of new process with the current organization. Various Agile adoption frameworks have proposed different checklists to test the readiness of an organization for becoming Agile, or to identify the required adaptation criteria. Transitioning to Agile, as a significant organizational initiative, is a strategic decision, which should be made with respect to key objectives of the target organization. Having a reliable anticipation of *how* a new process model will impact the strategic objectives helps organizational managers to choose a process model, which brings optimum advantage to the organization.

This thesis introduces a framework for evaluating new Agile practices (compartments of Agile methods) prior to their adoption in an organization. The framework has two distinguishing characteristics: first, it acts strategically, as it puts the strategic model of organization at the center of many decision makings that should be performed during Agile adoption; and second, it is based on a repository of Agile practices that allows the framework to benefit from the empirical knowledge of Agile methods, in order to improve the reliability of its outcomes. This repository has been populated through an extensive literature review of empirical studies on Agile methods.

The framework was put in practice in an industrial case, at one of the R&D units of Ericsson Company in Italy. The target R&D unit was proposed with a number of Agile practices. The application of framework helped R&D unit managers to strategically decide on the new process proposal, by having a better understanding of its strategic shortcomings and strengths. A key portion of framework's analysis results were evaluated one year after the R&D unit made the transition to Agile, showing that over 75% of pre-adoption analysis results came to reality after the enactment of new process into the organization.

Acknowledgments

I would like appreciate the professional and friendly support of my supervisor, Professor Eric Yu. He introduced me to the field of conceptual modeling, and in particular, goal-oriented analysis and design. Throughout my 5 years of research and study under his supervision, he has been always a rich source of inspiration for emerging ideas, and a great guide in elaborating new thoughts. I also want to thank my supervisory committee, Professor John Mylopoulos and Professor Steve Easterbrook, for their motivating support at various stages of my work.

I would like to thank all my colleagues who helped me throughout the graduate studies. In particular, I should thanks Jordi Cabot, Jorge Aranda, Golnaz Elahi, Jennifer Horkoff, Jocelyn Simmonds, Rick Salay, Jonathan Lung, Elizabeth Patitsas, Reza Manbachi, Neil Ernst, Michalis Famelis, Alicia Grubb. More specifically I want to appreciate the efforts of undergraduate students William Chang, Jing Tan, and Rutwij Devashrayee who helped me in some programming work.

I should thank my Mother and Father, who patiently bearing the pain of their older son being thousands of miles away, but never raised the issue, may I hesitate in the chosen path. I should also thank my in-laws who have been always supportive to my family. May Allah keep them all safe, happy, and successful. Last but not least, I want to thank my wife, Zeynab, who has been my partner of happiness and sadness, smile and tears, richness and poverty. The one who brought my daughter to this world – Fatemeh, who is the greatest blessing of Allah to me, and at the time of writing this thesis is just four month old.

Table of Contents

| | |
|--|-----------|
| Acknowledgments | iv |
| Table of Figures | xi |
| Chapter 1. Introduction | 1 |
| 1.1 Problem Statement..... | 3 |
| 1.2 Approach: Strategic Agile Pre-Adoption Analysis..... | 5 |
| 1.3 Contributions | 5 |
| 1.3.1 Publications and Their Mapping to the Thesis Structure | 6 |
| 1.3.2 Industrial experience | 7 |
| 1.3.3 Application of the Framework in areas other than Agile adoption..... | 8 |
| 1.4 Structure of This Thesis | 8 |
| Chapter 2. Background and Related Work | 11 |
| 2.1 Agile Method Adoption Frameworks..... | 11 |
| 2.2 Goal-Oriented Modeling..... | 13 |
| 2.2.1 Goal-Oriented Modeling Frameworks..... | 14 |
| 2.2.2 Goal Oriented Modeling Techniques..... | 26 |
| 2.3 Balanced Scorecards for Strategic Analysis | 29 |
| 2.4 Change Impact Analysis | 32 |

| | | |
|---|--|-----------|
| 2.5 | Current Approaches of Identifying Process Concerns..... | 33 |
| 2.5.1 | Process Assessment..... | 33 |
| 2.5.2 | Process Concern Identification in Small- to Medium-Size Organizations..... | 34 |
| Chapter 3. A Framework for Pre-Adoption Analysis of Agile Practices..... | | 35 |
| 3.1 | SAAF Rationale..... | 36 |
| 3.2 | Organization Strategic Model | 36 |
| 3.3 | Evidence Based Repository of Agile Practices | 37 |
| 3.4 | SAAF Process..... | 38 |
| 3.4.1 | Initialization Phase..... | 38 |
| 3.4.2 | Strategic Agile Practices Analysis..... | 39 |
| 3.4.3 | Strategic Actor Analysis | 41 |
| Chapter 4. Evidence Based Repository of Agile Practices | | 44 |
| 4.1 | Current Repositories of Agile practices..... | 46 |
| 4.2 | Evidence-Based Software Engineering | 46 |
| 4.3 | Building the Evidence-Based Repository of Agile Practices..... | 46 |
| 4.3.1 | Questions of the Systematic Review | 47 |
| 4.3.2 | Study Selection | 48 |
| 4.3.3 | Data Extraction | 48 |
| 4.3.4 | Data Analysis and Aggregation..... | 49 |
| 4.3.5 | Data Visualization | 50 |

| | | |
|-------------------|---|-----------|
| 4.4 | Results..... | 52 |
| 4.4.1 | Agile Practice Specific Objectives Dataset..... | 54 |
| 4.4.2 | Agile Practice Specific Requisites Dataset | 55 |
| 4.4.3 | Agile practice Visualization..... | 56 |
| 4.4.4 | Agile practices: Generic Snapshots | 57 |
| 4.4.5 | Catalogue of Agile Objectives | 60 |
| Chapter 5. | Strategic Analysis of Agile Practices | 62 |
| 5.1 | Organization Strategic Model | 62 |
| 5.1.1 | Strategies Graph..... | 63 |
| 5.1.2 | Further Elements of Strategic Model..... | 65 |
| 5.1.3 | Validation of Strategies Graph..... | 67 |
| 5.2 | Strategic Contribution Analysis | 68 |
| 5.2.1 | How to conduct the Strategic Contribution Analysis..... | 70 |
| 5.3 | Propagative Strategic Analysis..... | 72 |
| 5.3.1 | How to run the Propagative Strategic Analysis..... | 73 |
| 5.4 | Aggregative Strategic Analysis | 77 |
| 5.4.1 | How to run Aggregative Strategic Analysis..... | 78 |
| 5.5 | Strategic Balance Analysis..... | 78 |
| 5.5.1 | Balance Improvement..... | 79 |
| 5.5.2 | Balance Preservation | 80 |
| 5.5.3 | Balance Preservation across Categories..... | 82 |

| | | |
|--|---|------------|
| 5.5.4 | Homogenous Contributions across Categories..... | 83 |
| | | |
| Chapter 6. Strategic Actor analysis: an Approach for Identifying Process Concerns | | 86 |
| 6.1 | Lightweight Process Modeling for Identify as-is Process Concerns..... | 87 |
| 6.2 | Strategic Dependency Models..... | 88 |
| 6.2.1 | Strategic Actor | 88 |
| 6.2.2 | Strategic Dependency Diagram..... | 89 |
| 6.2.3 | Itemized Strategic Dependency..... | 90 |
| 6.3 | Identifying process concerns using ISDs..... | 92 |
| 6.4 | What kinds of process concerns are identifiable with this approach..... | 94 |
| 6.5 | Analyzing the Identified Process Concerns | 94 |
| 6.5.1 | Strategic Process Concern Analysis | 95 |
| 6.5.2 | Categorization of IPCs:..... | 96 |
| 6.5.3 | Strategic Cause-Effect Analysis of IPCs: | 97 |
| 6.5.4 | Expected Results of Strategic Concern Analysis..... | 100 |
| | | |
| Chapter 7. SAAF in Practice – An Industrial Case Experience in Ericsson . | | 102 |
| 7.1 | Demographic Information..... | 102 |
| 7.2 | Timeline | 103 |
| 7.3 | Candidate Agile Practices | 104 |
| 7.3.1 | Scrum Team..... | 104 |
| 7.3.2 | Sprint Planning (SP)..... | 105 |

| | | |
|-------------------|---|------------|
| 7.3.3 | Daily Scrum Meeting (DSM)..... | 107 |
| 7.3.4 | Potential Shippable Product (PSP) & Short Release (SR) | 108 |
| 7.4 | Initial Agile Adoption Framework | 110 |
| 7.5 | Experiences with Setting up the Strategies Graph | 111 |
| 7.6 | Experiences with Candidate Agile Practices & Retrieval of Strategic Knowledge | 112 |
| 7.7 | Experiences with Strategic Analysis Process..... | 113 |
| 7.7.1 | Strategic Contribution Analysis (SCA) | 114 |
| 7.7.2 | Propagative Strategic Analysis (PSA) | 115 |
| 7.7.3 | Aggregated Strategic Analysis (ASA)..... | 115 |
| 7.7.4 | Strategic Balance Analysis (SBA)..... | 116 |
| 7.8 | Evaluation of Strategic Analysis Results | 119 |
| 7.8.1 | Evaluation Process | 119 |
| 7.8.2 | Evaluation Result | 120 |
| 7.8.3 | Discussion on Evaluation Results | 125 |
| 7.9 | Experience with Strategic Actor Analysis | 125 |
| 7.9.1 | Using Itemized Dependency Diagrams for Extracting Process Concerns..... | 126 |
| 7.9.2 | Analyzing the Extracted Process Concerns..... | 130 |
| 7.9.3 | Agile Practices Capability Analysis | 139 |
| 7.10 | Final Results of Applying SAAF in Ericsson project | 142 |
| Chapter 8. | Conclusion and Future Work | 144 |

| | | |
|-----|--|------------|
| 8.1 | Limitations and Threats to Validity of SAAF | 147 |
| 8.2 | Future Work | 149 |
| | Bibliography | 153 |
| | Appendix 1. SAAF Evaluation Questionnaire | 160 |

Table of Figures

| | |
|--|----|
| Figure 1: EKD Goal Metamodel (Kavakli & Loucopoulos, 1999)..... | 16 |
| Figure 2: Conceptual goal metamodel of KAOS | 17 |
| Figure 3: part of <i>i*</i> metamodel..... | 19 |
| Figure 4: Strategic Dependency Model of a typical Software Process, adopted from (Yu & Mylopoulos, 1994) | 20 |
| Figure 5: Tropos goal metamodel (Susi <i>et al.</i> , 2005)..... | 22 |
| Figure 6: metamodel of AGORA goal graph (Kaiya <i>et al.</i> , 2002)..... | 27 |
| Figure 7: (Left) Four perspectives of Balanced Scorecard (Kaplan & Norton, 1996b); (right) A framework for Building BSC [source of images: Balanced Scorecard Institute® website]..... | 30 |
| Figure 8: Building Blocks of the SAAF | 35 |
| Figure 9: A portion of a sample strategies graph | 37 |
| Figure 10: Phase One: Initialization..... | 38 |
| Figure 11: Phase Two: Strategic Analysis of Candidate Agile Practices..... | 40 |
| Figure 12: Phase Three: Strategic Actor Analysis: Identifying as-is Process Concerns..... | 42 |
| Figure 13: Metamodel of Repository | 53 |
| Figure 14: Visualization of the Agile Practice “Pair Programming” for a Subset of its Objectives and Prerequisites | 57 |
| Figure 15: A Sample Strategies Graph..... | 64 |
| Figure 16: (Step 3) visualizing the Type and Rationale of the Contributions of the Candidate Agile Practice (CAP) "Sprint Planning" over a portion of the Strategies Graph (Example from the Ericsson’s case)..... | 70 |
| Figure 17: Metamodel of the contribution of Agile practices over strategic objectives, and the structure of the SG..... | 72 |
| Figure 18: Propagative Strategic Analysis: (left) original direct contribution; (right) propagated indirect contribution | 72 |
| Figure 19: Propagative Strategic Analysis; a conflicting scenario | 73 |

| | |
|--|-----|
| Figure 20: Propagation Examples | 75 |
| Figure 21: Sample i* Strategic Dependency Model..... | 89 |
| Figure 22: Part of Functional ISD, developed in Ericsson project | 91 |
| Figure 23: Part of Quality ISD, developed for R&D UNIT project..... | 92 |
| Figure 24: Major activities and artifacts of concern analysis | 96 |
| Figure 25: (Left) Sample Area of Concern, its Root Causes (RC), Intermediate Causes (IC), and Affected Strategic Goals; (Right) the cause-effect model of mentioned process concerns | 97 |
| Figure 26: Visualization of process concerns over Strategies Graph | 99 |
| Figure 29: Agile Adoption timeline in Target Organization | 103 |
| Figure 27: initial solution of the R&D unit for adopting Agile (Szalvay et al., 2008) | 110 |
| Figure 28: A portion of the Strategies Graph (SG) developed in the Ericsson case..... | 112 |
| Figure 30: overall distribution of the Number and the percentage of agreement types | 122 |
| Figure 31: Distribution of the Number and the Percentage of Agreement Types for Each Candidate Agile Practice | 123 |
| Figure 32: initial Fisd of Solution and System Manaher roles | 127 |
| Figure 33: Sample fisd, developed after the FIRST ROUND of interviews | 128 |
| Figure 34: SAmple evaluated QISD..... | 129 |
| Figure 35: strategic analysis of process concerns of category "documentation Problems" | 137 |

Chapter 1. Introduction

More and more organizations are considering adopting Agile as their software development processes (Ketter *et al.*, 2009; Gill, 2011). Agile methods have become popular partly due to their inherent characteristics such as simplicity or efficiency, and partly due to the limitations of traditional software processes such as ineffective feedback loop (Boehm & Turner, 2003; Cohn & Ford, 2003). Although Agile – from one perspective – is aiming at rapid software development, the transitioning to Agile is not often happening very quickly or without side-effects. In fact, Agile adoption is a significant organizational initiative, which must be carried out with respect to the key characteristics of an organization, as well as the attributes of ongoing projects (Qumer & Henderson-Sellers, 2008).

In management, it is acknowledged that key decisions of an organization must be in line with the overall organizational strategies (Kaplan & Norton, 2004). A clear and well-established strategic plan guides decision makers in the *evaluation* and the *selection* of various proposals, which would affect the organization in any scale (Krajewski *et al.*, 2009). Changing the software development process is recognized to be one the most challenging initiatives that any software organization might face (Boehm & Turner, 2005). It often requires a considerable amount of investment, mainly in terms of training human resources, and purchasing new tools. Therefore, from the management perspective, a newly proposed software process should meet the strategic concerns of the organization, and justifies its enactment costs (Slaughter *et al.*, 2006).

The need for a new process often becomes explicit when the inefficiency symptoms of the current process become sensible. The new process can be attained through different approaches, including *Situational Method Engineering* (Jansen *et al.*, 2007), systematic frameworks of *Software Process Improvement* (McFeeley, 1996; Niazi, 2006; SEI, 2009), predefined methods such as XP or Scrum (Schwaber, 1995; Beck *et al.*, 2001), or even ad-hoc

process models locally shaped in software companies. Regardless of the way that the new process is designed, it is expected to support strategic objectives of organization, does not cause unexpected side-effects, and removes the problems of current process model.

When a software organization decides on the adoption of a new development process, it should have a clear understanding of the strategic impacts of the initiative, and select a process model, which best supports its strategic goals. However, gaining such understanding, before the adoption of new process, is inherently complex, as such effects would often emerge after the enactment of the new process. The best case scenario is when the new process fully supports the strategic objectives, and matches the organization situation. But, this is not always the case and in many situations, the observed results of a new process were quite below its expected outcomes (Chiniforooshan Esfahani & Yu, 2010).

Recognizing the intensity and the diversity of the strategic effects of changing the development process of a software organization, this thesis addresses the process of transitioning to Agile processes from the perspective of organizational strategic management. The thesis defines a framework, called *Strategic Agile pre-Adoption analysis Framework* (SAAF), which is aimed at the *pre-adoption* analysis of candidate Agile practices, in order to evaluate their potential future impacts on the organizational strategic objectives, and their capability in resolving the problems of current process. To this end, the framework proposes a set of techniques for analyzing candidate Agile practices, and a set of components which support this analysis. By performing this analysis before enacting any new practices, one can anticipate potential mismatches between organizational strategies and candidate practices.

SAAF takes advantage of the empirical studies on Agile methods to improve the reliability of its analysis results. The flourishing body of empirical studies on Agile methods was a promising motivation to improve the way software organizations adopt Agile. The basic was to reuse the experiences of other organizations, while deciding upon the adaptability of an Agile method with a new organization. The fact that most Agile methods are built out of the assembly of various Agile practices, motivated us to build the *Evidence-Based Repository of Agile Practices*, as one of the major components of the SAAF (Chiniforooshan Esfahani & Yu, 2010). The repository supplies situational evidences for major Agile practices, and its content has been collected through systematic reviewing of empirical studies on Agile methods.

A number of frameworks have been proposed to provide guidance for transitioning to Agile (Sidky *et al.*, 2007; Qumer & Henderson-Sellers, 2008; Sureshchandra & Shrinivasavadhani, 2008; Krasteva *et al.*, 2010), but none takes a strategic perspective to link business goals to the selection of Agile practices. Indeed, the focus of current Agile adoption frameworks is mainly on the post-adoption activities, and in most cases very little attention has been made to the pre-adoption analysis activities. In many cases software organizations adopt Agile through the trial of new method in a pilot project (Szalvay *et al.*, 2008). However, this approach is not reliable as it cannot reveal the large-scale impacts of Agile adoption. A comprehensive overview of current state of the art in Agile adoption frameworks is presented in the Chapter 4 of the thesis.

The primary reasons which motivated the design of SAAF are as follows:

- The increasing tendency of software companies in adopting Agile, and the issues of ad-hoc approaches in designing custom Agile processes.
- The increasing number of empirical studies on Agile processes, and the lack of a systematic approach for accessing and reusing this body of knowledge towards low-risk adoption of Agile processes.
- The shortcomings of current Agile adoption frameworks in pre-adoption analysis activities, which are aimed at selecting right set of Agile practices.
- The significance of acting strategically in transitioning to a new process.

1.1 Problem Statement

One of the common concerns of project managers is to ensure that their development processes fit well with the particular needs of their projects. Despite the proposal of elaborate frameworks for building situational methods (Ralyté & Rolland, 2001; Saeki, 2003; Aydin *et al.*, 2005; Jansen *et al.*, 2007), many software companies still follow ad-hoc methods of software development, which are built intuitively by adopting some fragments from different methodologies and tailoring them to their development method (Henderson-Sellers, 2003; Bajec *et al.*, 2007). With this approach, the best case scenario is that the company will benefit from all advocated advantages of selected Agile practices. Unfortunately, this is not always the

case, and there are numerous reports of project failure that were due to the improper choice of development method (Slaughter *et al.*, 2006).

When an organization rethinks its software process, it is typically motivated by technical considerations, such as improving code quality, also by business and organizational concerns such as faster time-to-market, better responsiveness to client needs, and higher business competitiveness. A wide array of Agile practices are advocated by the various Agile methods; however, each practice may contribute to some strategic goals but not to others. This would add another layer of complexity to the choice of an Agile method for an organization (Rossi *et al.*, 2004). From the management perspective, in order to make an effective and successful transition to Agile, an organization needs to select a combination of practices that are best aligned with its strategic goals, while being mutually compatible and not conflicting

For instance, suppose that a software organization wants to incorporate “Pair Programming” (Beck *et al.*, 2001) as part of its development method. It is known that pair programming usually helps to achieve some objectives such as “reduced defect rate” and “real-time knowledge transfer”. But, are these objectives fully achievable in all project situations? Could other Agile practices, when combined with pair programming, facilitate (or hamper) these objectives? How would pair programming affect other local strategic objectives of the organization? Do we know all of the capabilities, pitfalls, and requirements of this Agile practice?

There are increasing empirical studies that have investigated the success or the failure of different Agile practices in different project situations. For instance, there exist over 100 empirical studies about “Pair Programming” in various situations¹. This large number of empirical studies on various Agile practices provides a baseline for designing a solution that can systematically make this evidence-based information available to practitioners, and help them during the process of method construction or evaluation.

Considering the significance of software processes, the dominant role of Agile methods among different categories of software processes, and the wide range of Agile processes that are built out of various Agile practice, this thesis is targeting the following problems:

¹ The search was run on libraries of IEEE Explorer, Springer, and Elsevier in January 2012

- How to anticipate the strategic impacts of transitioning to a new Agile process before the actual enactment of the process? This would aim at clarifying the organization-specific implications of transitioning to Agile, and trying to select a method which will best serve the strategic objectives of an organization.
- How to identify the capabilities of a proposed agile process in resolving current process problems? This includes a number of steps, such as systematic extraction of current process problems; root-cause as well as strategic analysis of extracted problems; and finally capability analysis of candidate process in resolving the problems.
- How to facilitate the public access to the situational benefits and side effects of various Agile practices? This involves, first, the identification of such situational attributes of agile practices, and second, the provision of a solution which facilitates this access.
- How to improve the reliability of process evaluation results at the pre-adoption stage, in order to gain further confidence in the enactment of the selected process?

1.2 Approach: Strategic Agile Pre-Adoption Analysis

In response to the above-mentioned problems, this thesis proposes the *Strategic Agile pre-Adoption analysis Framework* (SAAF) for the evaluation of a candidate Agile method, prior to its enactment within the development environment of an organization. It is intended to help project/process manager to decide whether a candidate Agile method is sufficient for resolving current process concerns, and is in conformance with the strategic objectives of the organization. Following a method engineering viewpoint (Jansen *et al.*, 2007), SAAF views any Agile method as a set of Agile practices (e.g., pair programming, planning game, daily meeting) aimed at achieving a set of Agile values (e.g., improved communication, collaboration, effectiveness). The framework is introduced in Chapter 3.

1.3 Contributions

The contribution of this thesis to the Computer Science has been in the form of a number of papers, published in and submitted to reputable Conferences and Journals; The use of proposed

framework for changing the development process in one industrial experience at Ericsson Company; and the introduction of a decision making framework, applicable for the evaluation of a wide range of organizational initiatives.

1.3.1 Publications and Their Mapping to the Thesis Structure

The following papers have been published along the period of this PhD research:

- H.Chiniforooshan, E.Yu, John Carbrey. “Extracting Contextual Complexities of Adopting Agile Practice(s)”, European System, Software & Service Process Improvement & Innovation (EuroSPI), Austria, Vienna (2012)
- H.Chiniforooshan, E.Yu, M.C.Annosi. "Strategically Balanced Process Adoption", International Conference on Software and Systems Process (ICSSP'11), Honolulu, USA, 2011.
 - The 5th chapter of this thesis is partly based on the ICSSP'11 paper
- T.Birkhölzer, H.Chiniforooshan, C.Dickmann, J.Vaupel, Stefan Ast, "Goal-Driven Evaluation of Process Fragments Using Weighted Dependency Graphs", International Conference on Software and Systems Process (ICSSP), USA, 2011.
- H.Chiniforooshan, E.Yu, M.C.Annosi. “Towards the Strategic Analysis of Agile Practices”, CAiSE'11 Forum, 23rd International Conference on Advanced Information Systems Engineering, London, 2011.
 - CAiSE'11 paper contains a brief summary of the SAAF – similar to the 3rd chapter of the thesis.
- H.Chiniforooshan, E.Yu, J.Cabot. "Situational Evaluation of Method Fragments: an Evidence-Based Goal-Oriented Approach", 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10), Tunisia, 2010. (Acceptance Rate: 11.3%)
 - CAiSE'10 paper introduced a process analysis technique, which its improved version is described in 5th chapter of the thises.

- H.Chiniforooshan, J.Cabot, E.Yu. "Adopting Agile Methods: Can Goal-Oriented Social Modeling Help?", 4th International Conference on Research Challenges in Information Science (RCIS'10), France, IEEE Publication, 2010.
- H.Chiniforooshan, E.Yu. "A Repository of Agile Method Fragments", International Conference of Software Process (ICSP'10), Germany, 2010.
 - 4th chapter of the thesis is partly based on the ICSP'10 paper
- H.Chiniforooshan, E.Yu, M.C.Annosi. "Itemized Strategic Dependency: a Variant of the *i** SD Model to Facilitate Knowledge Elicitation", 4th International *i** Workshop, Tunisia, 2010.
 - The *i**'10 paper is used in the 6th chapter of the thesis
- H.Chiniforooshan, E.Yu, M.C.Annosi. "Capitalizing on Empirical Knowledge during Agile Adoption", Research-on-Progress Workshop of Agile IEEE Conference, USA, 2010.
- H.Chiniforooshan, E.Yu. "Strategic Analysis of Agile Practices", to be submitted to the Journal of systems and Software.
- H.Chiniforooshan, E.Yu. "Identifying and Analyzing Strategic Process Concerns", to be submitted to the Journal of Software Process: Improvement and Practice.

1.3.2 Industrial experience

The SAAF has been tried out in one of the R&D units of Ericsson Company in Italy. The R&D unit was motivated to adopt an Agile process, in response to some inefficiency symptoms warning unit managers about their software development process. The pre-adoption analysis procedures of SAAF helped the R&D managers to investigate the potentially positive and negative impacts of candidate Agile practices on their strategic goals. As a result they identified some customizations needed to adapt candidate practices for the organization, as well as organizational changes needed to prepare for the enactment of those practices. The analysis results also helped middle managers overcome their doubts and hesitations in adopting new practices.

1.3.3 Application of the Framework in areas other than Agile adoption

One of the major contributions of SAAF is the introduction of a resourceful repository of Agile practices, which can contribute to various research questions in the area of software process. This repository has been well received and appreciated by the research community of software processes, as it was first introduced in the International Conference on Software Processes, 2010, Germany. The repository is now being used by another research group in the University of Applied Sciences Konstanz, Germany, and initial publication is accepted at ICSSP'11 (Birkhölzer *et al.*, 2011). This repository is available on-line at www.ProcessExperience.org.

Although this framework was developed in the context of Agile adoption and transition, it provides a framework for strategic decision making, which can be used for other initiatives. The primary reason which bounded the introduction of SAAF to the Agile methods is its dependency to the repository of Agile practices (Chiniforooshan Esfahani & Yu, 2010), which mostly supports Agile practices. Provided the expansion of repository with the information of non-Agile practices, the very same process and components of SAAF can be applied for non-Agile processes, or processes which are composed of Agile and non-Agile method fragments. However, even without the repository support, the strategic analysis procedures described in this thesis can be applied for evaluating various software processes.

Apart from the application of SAAF for software processes, the proposed strategic decision making processes are applicable in a wide range of organizational initiatives. For instance, the basic approach of strategic analysis was deployed in an “Innovation Improvement” initiative, at the Ericsson Company. The purpose of this initiative was to improve the potential of organization in proposing innovative ideas, leading to the business and technological success of organization. The use of strategic analysis methods of SAAF helped the organization to, first, come up with a clear and comprehensive set of innovation objectives; second, clarify the implicit relations of innovation improvement with the overall strategic objectives of organization; and third, evaluate the impact of alternative action plans of innovation improvement.

1.4 Structure of This Thesis

This thesis is organized as follows:

- **Chapter 2 – Background and Related Work**
 - This chapter describes the background and related work. It presents an overview of key approaches and frameworks of Agile adoption. This chapter also reviews the fundamentals of organizational strategic management, focusing on Balanced Scorecards. A thorough review of goal-oriented modeling approaches is presented in this chapter, as they form the basis of this work in model-driven strategic analysis.

- **Chapter 3 – A Framework for Pre-Adoption of Agile Practices**
 - This chapter introduces the SAAF, and described the main components of the SAAF, including a process model which connects various activities and artifacts of the framework.

- **Chapter 4 – Evidence Based Repository of Agile Practices**
 - This chapter describes the evidence based repository of Agile practices, which is one of the key components of the SAAF. This repository provides a systematic approach for accessing a huge body of evidential knowledge about various Agile practices. Although the repository is introduced as a SAAF component, it can be separately used, as a resourceful knowledge base of Agile practices.

- **Chapter 5 – Strategic Analysis of Agile Practices**
 - This chapter describes how to build the organization strategic model, and introduces the concepts of Strategies Graph (SG), which is one of the key components of the framework. The concept of organization strategic model has been inspired by the Balanced Scorecard's approach in strategic management of organizations (Kaplan & Norton, 1996a), but the visualization and analysis of

strategic objectives have been improved by Goal-Oriented modeling techniques of the i* framework (Yu, 1997). This chapter also describes the procedures, which will be used for analyzing every candidate Agile practices, based on the strategies graph of organization.

- **Chapter 6 – Strategic Actor analysis: an Approach for Identifying Process Concerns**
 - This chapter introduces the SAAF approach in identifying problems of current process. The proposed solution is based on the concept of strategic actors of organizations, and their dependency relations.

- **Chapter 7 – SAAF in Practice – the Experience Case in Ericsson**
 - This chapter describes the experience of SAAF in an industrial setting, which was performed in of the R&D units of Ericsson Company in Italy. The chapter provides demographic information of the organization, its situation with respect to the as-is process, and the motivation of organization for adopting Agile. The partial results of trying SAAF in this organization will be used throughout the rest of thesis chapters as illustrative examples of the framework.

- **Chapter 8 – Summary and Future Work**
 - This chapter presents a quick summary of the overall framework, describe its limitations and threats to validity, and at the end envisions some future work.

Chapter 2. Background and Related Work

This chapter presents the background and related work of the thesis. It first discusses the current state-of-the-art in Agile adoption, and introduces the primary Agile adoption frameworks. Then it describes concepts in strategic analysis and modeling, which are forming the backbone of this thesis. It discusses Goal-Oriented Modeling Frameworks and Techniques; introduces the Balanced Scorecards; and describes some related work in Change Impact Analysis.

2.1 Agile Method Adoption Frameworks

So far a number of frameworks have been proposed for adopting Agile methods. Sidky *et al.* (2007) proposed an Agile measurement index, for evaluating the degree that an organization can become Agile. Based on this index, they defined an Agile adoption framework, which at the first step seeks to identify *discontinuing factors* – indicators that their existence in an organization proves the unreadiness of organization for adopting Agile. Then, the framework proceeds by assessing the project and organization capability for adopting Agile, and based on their degree of potential agility, proposes the most suitable set of Agile practices. One application of this framework in adopting Agile methods for the development of life-critical systems has been explained in (Sidky & Arthur, 2007). Despite the novelty of this framework in proposing a set of Agile practices with respect to the potential agility of an organization, there is no guarantee that the proposed practices be the most suitable set for an organization.

Qumer & Henderson-Sellers (2008) also proposed an Agile adoption framework, which is backed by an agility measurement technique. The adoption framework, called Agile Adoption and Improvement Model (AAIM), introduces six levels of Agile adoption: Agile Infancy, Agile Initial, Agile Realization, Agile Value, Agile Smart, and Agile Progress. At each level, the AAIM seeks to incorporate certain Agile principles (e.g. Speed, Flexibility, people-orientation, Leanness, etc.) into the organization. This framework uses the 4-DAT agility measurement technique, which provides specific metrics for quantitatively assessing the agility

level of a software process. Unlike (Sidky *et al.*, 2007), the AAIM does not explicitly propose Agile practices to an organization, rather it focuses on localization of Agile principles. However, the reliance of this framework on a quantitative measurement scale makes it vulnerable to correctness of accumulative numbers that are supposed to represent the agility degree of a process.

Krasteva *et al.* (2010) proposed an experience-based framework for adopting Agile practices, which is based on assembly-based situational method engineering. The significant characteristic of this framework is its emphasis on the use of organization's (as well as others) experiences in dealing with a set of Agile practices, when they come to be proposed to an organization.

Sureshchandra & Shrinivasavadhani (2008) proposed a framework for adopting Agile in distributed development. The framework is supported by a tool, based on (Boehm & Turner, 2003), for evaluating the degree of agility and formality that is needed for a project. The framework is composed of four stages: Evaluation, to determine the degree of distribution of the project; Inception, to form the distributed teams, and the way Agile practice can be incorporated into the development infrastructure; Transition, to enact the Agile practices into the development activities; and Steady State, to provide baselines for smoothly adding further distributed teams into the project.

Rohunen *et al.* (2010) argued that most of the current Agile adoption frameworks have three common characteristic features: (1) they are usually composed of stages; (2) each propose a measurement metric for evaluating the agility degree of organization or process; and (3) they provide mechanisms for managing the dependencies of Agile practices while they get incrementally incorporated into an organization.

A similar concept to the Agile adoption is studied under the Agile adaptation frameworks. Cao *et al.* (2009) proposed a framework for adapting Agile development methodologies, which is theoretically based on Adaptive Structuration Theory (Jones & Karsten, 2008) and multiple case studies. The framework investigates the adaptation of Agile methods with respect to the project and organization factors. Aydin *et al.* (2005) proposed two approaches for the adaptation of Agile method fragments with the organization context. In a more recent work (Mikulenias & Kapocius, 2011), acknowledging the fact that "the move toward Agile is often

hampered by the wealth of alternative practices”, proposed a technique for prioritizing Agile practices for their adaptation in an organization.

2.2 Goal-Oriented Modeling

Goal orientation in computer science dates back to the early 70’s, when it emerged as a problem solving strategy in artificial intelligence (Nilsson, 1971). Henceforth, goal-oriented approaches evolved in variety of CS disciplines, such as Enterprise Modeling (EM), Requirements Engineering (RE), and Business Process Management (BPM). The primary concern of goal-oriented approaches is to propose the intentional perspective of a subject modeling domain, via systematic analysis and representation of domain objectives and strategies (Lamsweerde & Letier, 2004). GO modeling has some reportedly referenced advantages, such as:

- Goal models provide the intentional view of modeling domain, which can express the motivations, intents and rationales behind activities (Yu & Mylopoulos, 1994)
- Goal models can be refined in order to represent complex structures of subject domain (Lamsweerde, 2001). Moreover refined goal models can reveal the sources of conflicts, and help in analyzing alternative approaches for conflict resolution.
- Goal models can provide a precise criterion for sufficient completeness of domain model (Lamsweerde, 2001). With respect to a goal model, an specification is complete if it satisfies all of the objectives, represented in the goal model (Yue, 1987).
- Goal models can guide the decision making process at various levels of an organization, as they represent the hierarchy of objectives (Harmsen & Saeki, 1996). Moreover, the process of goal analysis can help organizations in clarifying their objectives, yet defining organizational strategies.
- Goal models step further representing structural and behavioral views of subject domains, and capture non-functional aspects, such as qualitative attributes and qualifications. Moreover, goal models can capture social aspects of the modeling domain, through their built-in association to agents (Lamsweerde & Letier, 2004).

A number of definitions have been proposed for Goal. From RE perspective, Lamsweerde (2004) defines goals as “prescriptive intentions that should be satisfied by the corporation of software and environment agents”. From EM perspective, Kavakli & Loucopoulos (1999) define goals as “desired state of affairs that needs to be attained”. Moreover, they emphasize on the role of stakeholders in the design of goal models. In this regard, different goal-oriented frameworks have been proposed for development, representation, and analysis of goal models, with specific or generic application areas. The frameworks were then supported with variety of techniques, facilitating or enhancing their capabilities.

So far, most of the attempts that have been performed for GO modeling of SDMs were limited to the process aspect of methodologies. Besides, neither of currently existing GO process modeling approach was based on language-independent methodology (or process) metamodels. As mentioned before, Colette (1998) defined NATURE language, which had taken a goal-based approach for context-oriented representation of software processes. Later on, Colette (1998) proposed a process modeling notation, called map, for goal-based situational method engineering. However, a map can be considered as graphical representation of many of the concepts, which had been defined in NATURTE. On the other hands Yu & Mylopoulos (1994) highlighted the importance of understanding WHY in software processes, and proposed the application of a goal-oriented modeling framework, the i*-framework, for this purpose. The i*-framework has been also deployed by Cares *et al.* (2006) for quality-driven software process improvement.

This section presents a survey of major goal-oriented modeling frameworks, as well as their analysis with respect to the framework characteristics; and framework applicability for goal-oriented modeling of software development methodologies. Reviewed frameworks are: EKD, KAOS, i*, NFR framework, and Tropos. This section also presents the review and analysis of a number of goal-oriented techniques, including AGORA, GBRAM, VIVA, GOIG, PRiM, RSiD, and AGRA.

2.2.1 Goal-Oriented Modeling Frameworks

The reviewed goal-oriented modeling frameworks are analyzed with respect to the following criteria:

- Model Development: the primary process that the framework advocates for eliciting model elements, and developing the goal models.
- Model Representation: the main representation style that the framework suggests, i.e. Formal Semantic, Graphical Notation, or Informal Models
- Model Evaluation: the kind of analysis techniques provided for evaluating goal models, i.e. qualitative, or quantitative.
- Modeling Paradigm: the focal point of modeling, which can be Goal Oriented, Agent Oriented, or both.
- Modeling Objective: the primary purpose of framework for goal modeling.
- Tool Support: CASE tools that help deployment of the framework.

Moreover, the capability of each framework in goal-oriented representation of software development methodologies will be analyzed by investigating possible alignments between its goal modeling components, and the elements of SPEM 2.0 Method Content Package.

2.2.1.1 Enterprise Knowledge Development (EKD)

The EKD is an Enterprise Modeling (EM) framework, which advocates the use of goal models for expressing organizational objectives (Kavakli & Loucopoulos, 1999). This framework represents the enterprise knowledge with the help of six interrelated submodels: *Goal Model*, *Business Rule Model*, *Concept Model*, *Business Process Model*, *Actors and Resources Model*, and *Information System Model* (Persson & Stirna, 2001). Besides, the framework defines two modeling languages, one based on simple graphical notations and weak formality, and the other based on rich semantic and formality. However, the experience results of (Stirna & Persson, 2007) supports the superiority of simple-graphical, over formal EM approaches, while interacting with stakeholders.

The Goal Model of EKD hierarchically structures the enterprise objectives, and represents an intentional view of functional processes. It is just intended to answer WHY questions, regarding organizational objectives, and HOW, WHAT, and WHO questions are supposed to be handled in other submodels. Thus, as depicted in Figure 1, EKD goal metamodel does not

cover some of the basic organizational concepts, such as actor, activity, or resource. Instead, it uses *Inter-Model relationships*, for modeling the correlations that exist among the components of different submodels.

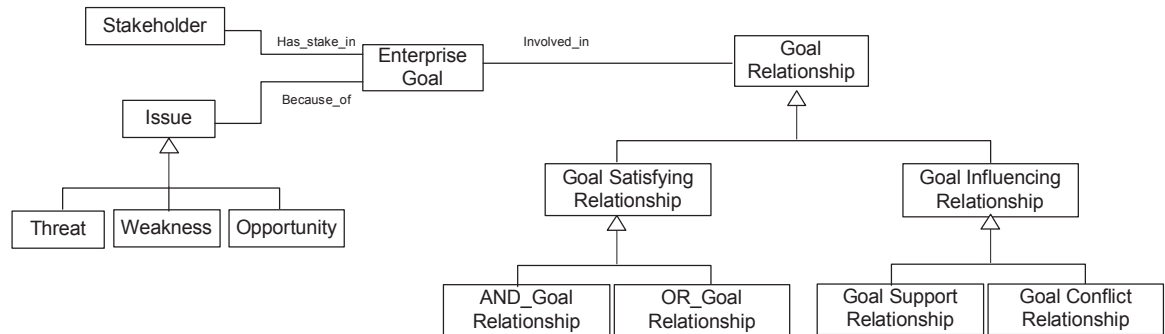


Figure 1: EKD Goal Metamodel (Kavakli & Loucopoulos, 1999)

EKD framework proposes a systematic process for model development, which has two characteristics: first, it emphasizes the importance of parallel development of submodels, and second, it stresses the effectiveness of participatory EM, in comparison with consultative approach. In participatory approach, the involvement of all stakeholders in modeling practices has been reported as a key factor in developing more complete models, and enhancing individuals' knowledge. However, in consultative approach, stakeholders are just viewed as information providers for analysts, and evaluators of final models (Stirna *et al.*, 2007). Table 1 summarizes the key characteristics of goal modeling in EKD framework.

As mentioned before, the goal submodel of EKD is just intended to represent the intentional aspect of an enterprise, and has no concern in modeling process or human aspects. Thus, the EKD goal submodel provides very restricted support for modeling method elements, defined in SPEM 2.0 Method Content Package. The only possible association that can be made is the use of EKD Goal notation, for representing method Qualifications. Table 2 suggests the possible alignments between EKD goal submodel and SPEM 2.0 Method Content Package.

2.2.1.2 Knowledge Acquisition in autOated Specification (KAOS)

The KAOS is a goal-directed requirements engineering framework, focused on requirements acquisition, formalization, and evaluation (Lamsweerde & Letier, 2004). It proposes a conceptual metamodel, which defines the primary elements of a goal oriented requirement model; and an acquisition strategy, which suggests how to elicit system and agents goals, step-by-step. Figure 2 shows the conceptual metamodel that KAOS deploys for goal-directed requirements acquisition.

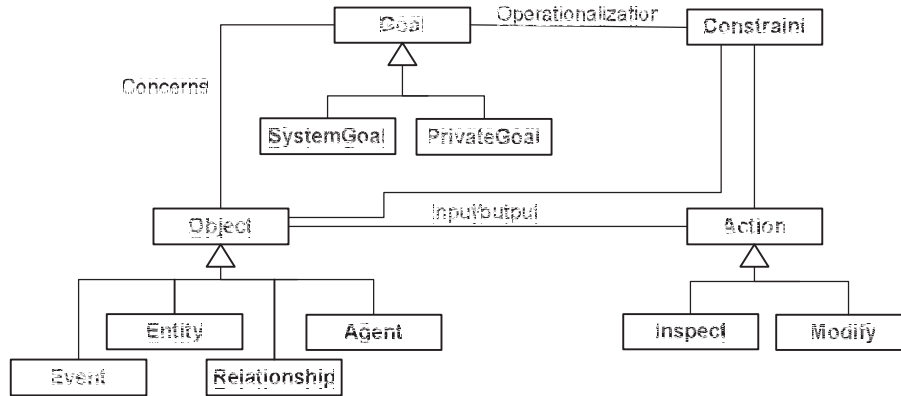


Figure 2: Conceptual goal metamodel of KAOS

The concept of Goal in KAOS is defined as “a nonoperational objective to be achieved by the composite system”. Thus, it should not formulate an object or activity of system agents. For instance, in a library system, a goal can be defined as “Achieve [BookRequestSatisfied]”, which means that the system should be able to achieve the state that satisfies any book request (Dardenne *et al.*, 1993). This objective is nonoperational as it does not formulate any specific activity of system agents. KAOS categorizes goals into SystemGoals and PrivateGoals, respectively referring to the objectives that should be achieved by the system, and agents. Moreover, KAOS defines five goal patterns: Achieve, Cease, Maintain, Avoid, and Optimize.

KAOS process follows a top-down strategy for goal acquisition. It starts by identifying high-level goals, then after identification of potential agents, decomposes the goals into operational objectives, which will be modeled as Constraints. The other steps of KAOS acquisition strategy include the identification of alternatives and the assignment of responsibilities to the agents. Letier & Lamsweerde (2004) presented a technique for evaluating goal satisfaction, both quantitatively and qualitatively. This technique refines goal

models with a probabilistic layer for reasoning about partial goal satisfaction. Darimont *et al.* (1997) developed a tool, called GRAIL, for goal-oriented modeling and specification in KAOS. Table 1 summarizes the key characteristics of goal modeling in KAOS framework.

KAOS can formally represent an intentional view of software development methodologies, as its basic concepts can cover most of the elements, defined in SPEM 2.0 Method Content Package. The concept of Agent can represent Role Definition, also Qualification and Responsibility Assignment as it specifies capabilities, activities, and expected knowledge of a role performer. The concept of Goal can represent the SPEM 2.0 Task Definition, referring to functional objectives of the method. Besides, the concept of Action can be used instead, if the specification of pre and post conditions be necessary. The concept of Entity can represent Work Product Definition and Tool Definition. Furthermore, from methodological perspective, KAOS systematically supports the development process from requirements to architecture (Lamsweerde, 2003). Table 2 suggests the possible alignments between KAOS framework and SPEM 2.0 Method Content Package.

2.2.1.3 *i Modeling Framework**

The *i** is an agent-oriented modeling framework, applicable for goal-oriented requirements engineering, software process modeling, business process engineering, and organizational impact analysis (Yu, 1997). Unlike other organizational modeling methods that focus on structural and behavioral aspects, *i** is intended to represent rationale aspects of an organization, which deal with Why questions (Yu & Mylopoulos, 1994). The *i** framework is composed of two modeling components: Strategic Dependency (SD) models, for describing the dependency relations among organizational actors; and Strategic Rationale (SR) modes, for representing actor objectives and their alternative ways of fulfillment, also the required resources.

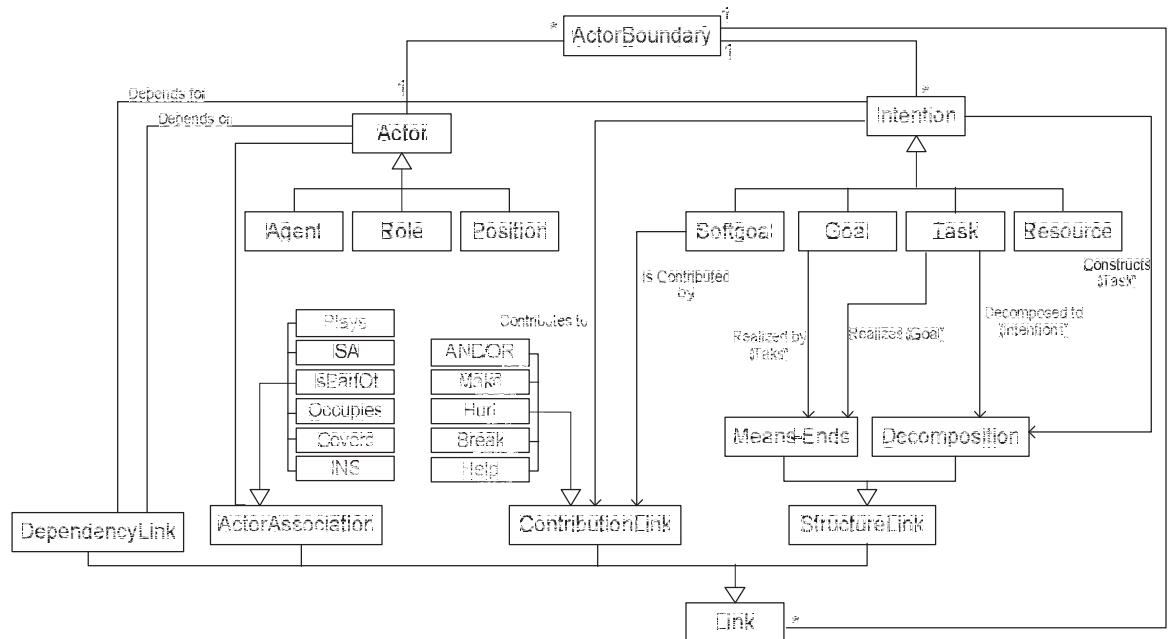


Figure 3: part of *i** metamodel

The *i** framework focuses on intentional actors, which have goals to achieve, tasks to perform, and resources to furnish (Yu, 1997). It introduces three types of actors: Agent, which represents a concrete actor, system or human, with specific intentions; Role, which is an abstract actor; and Position, which represents a set of roles typically assigned jointly to one agent. Actors are supposed to achieve their objectives either by themselves, or by depending on other actors. In *i**, actor dependency relations have been classified into four categories, based on the object of dependency (dependum): Goal, Softgoal, Task, and Resource. Dependency relations depicted in SD models are more abstract than those represented in SR models.

Strategic Rationale diagrams represent more elaborate view of actors' intentional elements: Goal, Softgoal, Task, and Resource. They encompass the internal view of each actor within an actor boundary, and specify the structural and contributitional relations of internal elements. In *i** framework, a Goal represents a state that is either achievable, or not; an Softgoal refers to a fuzzy state that its fulfillment status may range from fully achieved to fully denied. Each *i** Goal can be followed by a set of alternative Tasks, which act as means to that end. Moreover, *i** Tasks can be decomposed into other intentional elements.

Recently, an standard version of i^* framework has been published as part of a requirements engineering framework, *User Requirements Notation (URN)* (Z.151, 2008). URN deployed i^* framework as *Goal-oriented Requirements Language (GRL)*, in order to express business goals, quality attributes, and design alternatives (Amyot, 2003). Moreover, the i^* framework is supported by a number of goal-oriented techniques, e.g. PRiM (Grau *et al.*, 2008), for facilitating the process of model development. Table 1 summarizes the key characteristics of the i^* framework.

The i^* framework has been previously used for modeling organization specific software development processes (Yu & Mylopoulos, 1994). However, that work was mainly focused on the analysis of the dependency relations that exist among development team members, using SD models. i^* models have been also used for knowledge management in software development organizations (Elahi *et al.*, 2008). Figure 4 shows an example of representing strategic dependency relations in a typical software process. For instance, this model shows the dependency of *Customer* to the *Project Manager* for acquiring a Developed System, or the dependency of *Programmer* to the *Project Manager* for *Career Advancement*.

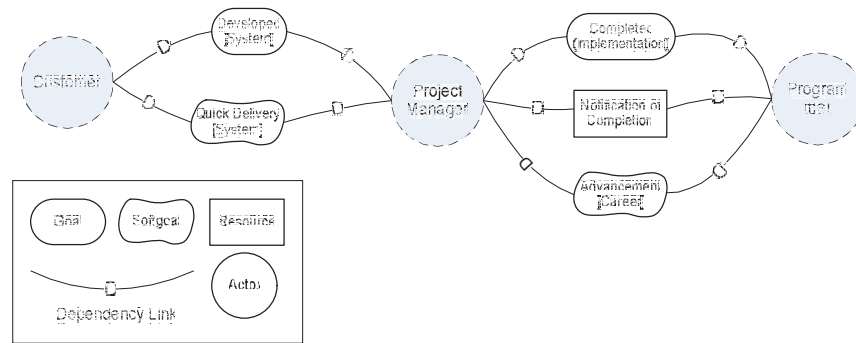


FIGURE 4: STRATEGIC DEPENDENCY MODEL OF A TYPICAL SOFTWARE PROCESS, ADOPTED FROM (YU & MYLOPOULOS, 1994)

The basic modeling concepts of i^* supports representation of primary elements of SPEM 2.0 Method Content Package. The i^* Role can be used for representing SPEM 2.0 Role Definition. The i^* Goal can be used for representing Task Definitions, as activities that should be performed. However, the practical representation of method tasks or their comprising steps, can be represented by i^* Task. The concept of i^* Softgoal can visualize Qualifications expected from actors, or activities. Moreover, the concept of Belief in i^* can help modeling method

Guidance. Table 2 suggests the possible alignments between i* modeling framework and SPEM 2.0 Method Content Package.

2.2.1.4 NFR Framework

The NFR Framework presents a systematic approach for addressing non-functional requirements (quality attributes), such as security, performance, accuracy, and flexibility in software projects (Chung *et al.*, 2000). This framework deals with identification, representation, categorization, and operationalization of software NFRs; and defines basic categories of non-functional requirements, which are generally applicable to development projects. NFR Framework is supported by a CASE tool, called NFR-Assist (Tran & Chung, 1999).

The primary artifact of this framework is Softgoal Interdependency Graph (SIG), which hierarchically visualizes softgoals, and specifies their structural (AND/OR) and contributinal (-, --, ?, +, ++) interdependencies. It also presents an evaluation procedure to determine the satisfaction degree of a given softgoal, in each design alternative. This framework introduces three types of softgoals: NFR Softgoals, for representing non-functional requirements; operationalizing softgoals, for modeling lower-level techniques for satisfying NFR softgoals; and Claim Softgoals, for justifying design rationales.

This framework provides a process-oriented approach for handling NFRs throughout a development project (Lapouchnian *et al.*, 2007). The approach focuses on design rationales that are tight to design alternatives, and through systematic evaluation of each alternative, tries to optimize the design decisions. For this purpose, NFR Framework classifies relevant knowledge of NFRs into three types of catalogues: NFR Type Catalogue that encompasses concepts related to an specific NFR, e.g. security; Method Catalogue that classifies various operationalization techniques that impact an NFR; and Correlation Rule Catalogue that holds implicit interdependencies among NFRs. Table 1 summarizes the key characteristics of the NFR framework.

As NFR framework is based on i* modeling framework, it contains most of the modeling concepts defined in i*. Thus, it should have inherited the capabilities of i* framework in modeling software development methodologies. However, the primary application of NFR framework is for the goal-oriented analysis of quality attributes, as supplementary part of

system development. Table 2 suggests the possible alignments between NFR framework and SPEM 2.0 Method Content Package.

2.2.1.5 Tropos

Tropos is a requirement-driven software development methodology, which is based on i* modeling framework (Castro *et al.*, 2001). Tropos methodology span in four phases: Early Requirements, which extracts use requirements in goal dependency and rationale diagrams; Late Requirements, which complements the early requirements by adding System as new actor; Architectural Design, which describes system components; and Detailed Design, which specifies the behaviour of architectural components.

Figure 5 shows the Tropos goal metamodel, which supports goal rational diagrams. Similar to i*, Tropos classifies goals into Hard Goals, dealing with functional requirements; and Soft Goals, representing quality attributes. The concept of Plan in Tropos corresponds to the i* Task. The primary difference of Tropos goal model with i* SR diagram is on possibility direct AND/OR decomposition of goals in Tropos.

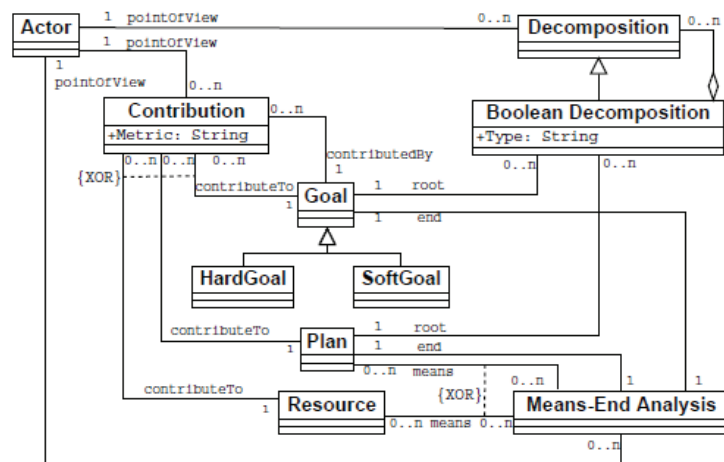


Figure 5: Tropos goal metamodel (Susi *et al.*, 2005)

(Gonzalez-Perez *et al.*, 2007) proposed a formal evaluation framework, which is based on formal representation of Tropos (Gonzalez-Perez *et al.*, 2007). This framework supports both forward goal analysis (starting from high-level goals, evaluating down to leaf nodes) and

backward goal analysis (starting from leaf nodes, evaluating the higher-level goals). Table 1 summarizes the key characteristics of the Tropos framework.

Similar to the NFR framework, Tropos also inherits modeling capabilities of i* for representing software development methodologies. However, the possibility of direct goal decomposition in Tropos goal models can facilitate the representation of method goals. Nevertheless, Tropos is a software development methodology by itself, and its modeling approach is counted as an extension to i*. Table 2 suggests the possible alignments between NFR framework and SPEM 2.0 Method Content Package.

Table 1 : Goal Oriented Modeling Frameworks Analysis

| | EKD | KAOS | i* | NFR | Tropos |
|-----------------------------|--|-------------------------------|--|---------------------------------------|---|
| Model Development | participatory approach | Systematic/ Top-Down Approach | Systematic (e.g. PRiM, RiSD) | Systematic, Category based | Evolutionary from early RE to detailed design |
| Model Representation | Graphical Notation / Formal semantic | Formal semantic | Graphical Notation | Graphical Notation | Graphical Notation / Formal Semantic |
| Model Evaluation | N/A | qualitative / quantitative | qualitative | qualitative | qualitative (forward/Backward) |
| Modeling Paradigm | Goal Oriented | Agent / Goal Oriented | Agent / Goal Oriented | Goal Oriented (focused on softgoals) | Agent Oriented |
| Modeling objective | Enterprise Modeling (intentional view) | Organization Modeling | Organization Modeling, Process Engineering | Software / System quality Engineering | Software Development |
| Tool Support | Generic Modeling Tools | GRAIL | OpenOME, REDEPEND | NFR-Assist | TAOM, OME, T-Tool, GR-Tool |

Table 2 : Goal Oriented Modeling Frameworks Alignment with SPEM 2.0 Method Content

Package

| | EKD | KAOS | i* | NFR | Tropos |
|----------------------------------|-------------|----------------------|----------------------------------|----------------------------------|----------------------------|
| Role Definition | Stakeholder | Agent | Role | N/A | Actor |
| Task Definition | N/A | Goal / Action | Goal / Task | Operationalizing Softgoal / Task | Hardgoal / Plan |
| Step | N/A | Action | Task | Operationalizing Softgoal / Task | Plan |
| Work Product Definition | N/A | Entity | Resource | N/A | Resource |
| Qualification | Goal | Agent / Private Goal | Softgoal | NFR Softgoal | Softgoal |
| Responsibility Assignment | N/A | Agent | Actor Boundary, Dependency Links | N/A | Actor Boundary, Dependency |
| Tool Definition | N/A | Entity | Resource | N/A | Resource |
| Work Product Relationship | N/A | N/A | N/A | N/A | N/A |
| Guidance | N/A | N/A | Belief | Claim Softgoal | Softgoals |

2.2.2 Goal Oriented Modeling Techniques

Different techniques have been proposed for supporting goal-oriented modeling. Some of these techniques were designed for specific frameworks, such as PRiM technique for i* framework; and many of them have generic application in goal modeling, e.g. AGORA. This sub-section presents a review of a number of goal-oriented modeling techniques, and then analyzes them based on the following characteristics:

- Base Framework
 - Generic for all frameworks
 - Specifically designed for a certain framework
- Purpose of Technique
 - Goal Elicitation
 - Goal Classification
 - Goal Elaboration
 - Goal Prioritization
 - Change Management
 - Conflict Management
 - Goal / Agent Assignment

2.2.2.1 *Attributed Goal-Oriented Requirements Analysis (AGORA)*

The AGORA is a technique that supports goal oriented requirements analysis (AGORA) methods (e.g. i*, KAOS, and GRL) with quantitative analysis power (Kaiya *et al.*, 2002). This technique can be applied on any goal model that advocates top-down structuring of goals, through AND/OR graphs. In AGORA, a goal graph is supported by a set of attribute value, which specify the contribution value of edges, and preference value of nodes. These attributes

can quantitatively help an analyst in recognizing conflicts, analyzing the impacts of requirements change, and deciding on alternatives. The metamodel of AGORA goal graph is depicted in Figure 6.

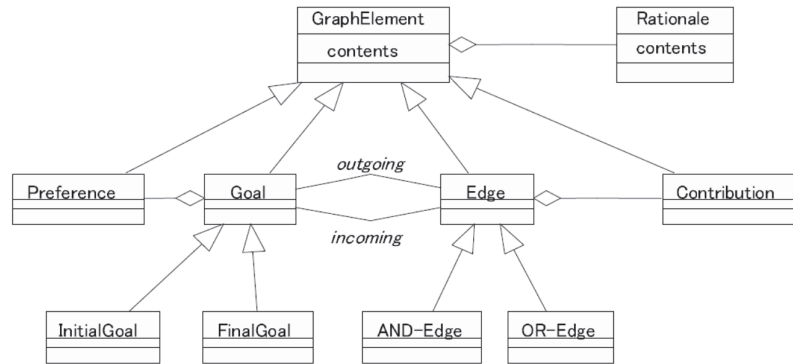


Figure 6: metamodel of AGORA goal graph (Kaiya *et al.*, 2002)

2.2.2.2 Goal-Based Requirements Analysis Method (GBRAM)

The GBRAM proposes a systematic method for Identification, refinement, and classification of goals (Annie, 1996). The activities of this method are classified in two groups: Goal Analysis, which deals with exploration, identification, and organization of goals; and Goal Refinement, which involves refinement, elaboration, and operationalization of goals (Antón & Potts, 1998). The GBRAM proposes a set of heuristics that help goal-oriented analysis. It highlights that stakeholders usually express their goals in terms of activities, and analysts are supposed to use systematic questions for identification and categorization of goals. This method uses schema representation for describing goals.

2.2.2.3 Visual Variability Analysis (VIVA)

(Gonzales-Baixauli *et al.*, 2004) proposes a goal analysis method, which is based on NFR Framework (Chung *et al.*, 2000). In the process of modeling, it first builds a goal model of functional requirements, as well as softgoal model of non-functional requirements (NFRs). Then, it evaluates functional nodes based on their quantitative contribution to the NFRs. The result of this analysis will be fed back to the initial goal models for refinement.

2.2.2.4 Goal-Oriented Idea Generation (GOIG)

Goal-Oriented Idea Generation (GOIG) is also a requirement elicitation method, which emphasizes the involvement of all stakeholders, for building more complete goal graphs (Oshiro *et al.*, 2003). It proposes an step-by-step method, based on controlled brainstorming, for goal identification, decomposition, and categorization.

2.2.2.5 PRiM

PRiM is a process reengineering method, which uses the i* Strategic Rationale models for process modeling and analysis (Grau *et al.*, 2008). This method is composed of six consequent phases that starts by analyzing, modeling, and reengineering the current process; and then after generating and evaluating the alternatives, ends with a new process specification. Here, we focus on the second phase of PRiM, which suggests an step-by-step method for constructing i*-based process models.

The first step of PRiM for constructing an i* model of a process, is the identification of actors. The second step, is to build an operational i* model, which includes the identification actor's primary activities, modeling them as top-goals, and refining them into tasks and resources. Following that, the third step of PRiM builds an intentional i* model, which complements the operational model. This step, iteratively adds the intentional elements, in order to represent the rationalities behind tasks and quality attributes that are expected. The last step, tests the validity of constructed i* model, by checking whether all of the intended process concepts (analyzed during the first phase of PRiM) has been mapped to the i* model.

2.2.2.6 RiSD

The RiSD is a methodology for constructing i* Strategic Dependency models (Grau *et al.*, 2005). The primary purpose of RiSD is to develop a socio-technical system model, which is traceable to its grounded social system model. Thus, during the first phase, RiSD develops an SD model of social system, by identifying actors, establishing goal dependencies, and then classifying and analyzing goal dependencies. Then, during the second phase, software system will be added to the social model, its subsystems will be identified, and the whole model will be refined. The focal point of RiSD is on iterativeness of its activities.

2.2.2.7 Agent Goal Responsibility Assignment (AGRA)

AGRA is a technique for refining goals, identifying agents, and exploring various ways of goals/agents assignment (Letier & Lamsweerde, 2002). This technique, which is built on top of KAOS framework, proposes a formal definition for the concept of realizability, and after introducing a taxonomy of goal realizability problems, defines a set of tactics for refining goals and assigning them to agents. Table 3 summarizes the characteristics of reviewed goal-oriented techniques.

Table 3: Capability Analysis of Goal-Oriented Techniques

| | Base Framework | Goal Elicitation | Goal Classification | Goal Elaboration | Goal Prioritization | Goal Evaluation | Change management | Conflict Management | Goal/Agent Assignment |
|--------------|---------------------|------------------|---------------------|------------------|---------------------|-----------------|-------------------|---------------------|-----------------------|
| AGORA | | | | | ✓ | ✓ | ✓ | ✓ | |
| GBRAM | | ✓ | ✓ | ✓ | | | | | |
| VIVA | NFR Framework | | | ✓ | | | | ✓ | |
| GOIG | | ✓ | ✓ | ✓ | | | | | |
| PRiM | <i>i*</i> Framework | ✓ | ✓ | ✓ | | | | | ✓ |
| RiSD | <i>i*</i> Framework | ✓ | | | | | | | ✓ |
| AGRA | KAOS | | | ✓ | | | | | ✓ |

2.3 Balanced Scorecards for Strategic Analysis

In 1990's, Kaplan and Norton introduced the concept Balanced Scorecard (BCS) in a number of articles in *Harvard Business Review*. The basic idea of BSC is to support the strategic management of organizations, through the structuring of portfolios' strategies at different levels. One of the major goals of BSC was to translate the often vague and high-level

organization vision and mission statements into actionable and measurable objectives that, first, all are related to the high level strategies of organization, and second, each can be understood and implemented by the accountable staff of organization (Kaplan & Norton, 1996a).

The BSC suggests that organization strategies should be viewed from four perspectives: (1) Financial –strategies for profitability of an organization from shareholders’ viewpoint; (2) Customer – strategies for creating value from customers’ perspective, (3) Internal business Process – strategies for improving the key business processes of organization; (4) Learning & Growth – strategies for improving the qualifications of individuals and the growth of corporate. As shown in Figure 7 (right), the high-level strategies of an organization (which are often articulated as vision and mission statements) should be decomposed into actionable Objectives, categorized by the four perspectives, and attributed by quantifiable Measures, desired Targets, and list of supporting Initiatives (Kaplan & Norton, 1996b).

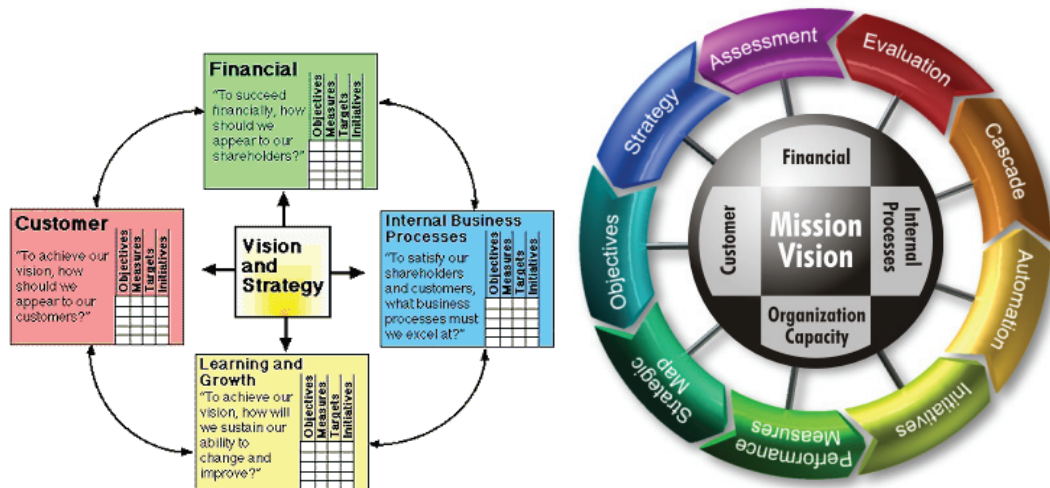


FIGURE 7: (LEFT) FOUR PERSPECTIVES OF BALANCED SCORECARD (KAPLAN & NORTON, 1996B); (RIGHT) A FRAMEWORK FOR BUILDING BSC [SOURCE OF IMAGES: BALANCED SCORECARD INSTITUTE® WEBSITE]

The initial concepts of BSC were then supported by detailed frameworks for building and implementing balanced scorecards. *Strategy Maps* (Kaplan & Norton, 2004) was proposed as for linking strategic objectives among four perspectives, and clarifying their cause-effect relation. The *Balanced Scorecard Institute*® proposed a framework (called, *Nine Steps to*

SuccessTM) for strategic planning and management, which is based on BSC key concepts, and is composed of nine steps:

1. *Assessment* – to assess the vision, mission, enablers, and values of organization. Also to prepare a change management plan, identify key participants, and communication channels.
2. *Strategy* – to develop the elements of organization’s strategy (strategic results, themes, and perspective), through participatory workshops.
3. *Objectives* – to translate the strategic elements of first two steps into strategic objectives, which are actionable and measurable components.
4. *Strategy Maps* – to build one enterprise-wide strategy map
5. *Performance Measures* – to identify leading and lagging measures of each strategic objective, as well as their expected targets, thresholds, and baselines.
6. *Initiatives* – to develop initiatives that support strategic objectives, and specifying the responsible staff.
7. *Automation* – to apply the performance measurement software for collecting and reporting the performance information.
8. *Cascade* – to achieve organization alignment around the strategy. An Enterprise-level scorecard is cascaded to business- and support-level, and then to team- and individuals-level scorecards. i.e. translating high-level strategies into lower level objectives, actions, and measures.
9. *Evaluation* – to evaluate the completed scorecard, by asking questions such as “Are our strategies right?”, “Are the performance measures correct?”

Balanced Scorecard Institute® also proposed a maturity model for assessing the quality of strategic management (SMG, 2010). This model defines eight dimensions for strategic management and proposes five levels of maturity for each dimension.

BSC has been also investigated by Computer Science researchers, and shown successful experiences in the software industry. Martinsons *et al.* (1999) proposed a customized BSC for Information Systems (IS), in which the original BSC perspectives are substituted by: *business value*, *user orientation*, *internal process*, and *future readiness*. They argued that the strategic goals and activities of an IS organization can be classified into these perspectives, and proposed a number of measures for evaluating each category. Following the BSC guidelines

(Kaplan & Norton, 1996a), they stated that all of the metrics should be quantifiable, easy to understand, and have meaningful analysis results. Grembergen & Amelinckx (2002) also proposed a customized BSC for e-Business, which consists of four perspectives: *customer-orientation*, *business contribution*, *operational excellence*, and *future orientation*. Each perspective is then supported by a number of quantifiable measures.

There are numerous reports of using BSC for strategic management in the software industry. (Huang & Hu, 2007) reported the successful application of BSC for strategic alignment of IT and business departments of an enterprise. They stated that BSC helped the enterprise's integrated strategic-planning and internal communication. Kim *et al.* (2003) explained a application of BSC (a customized instance of BCS from customer perspective) for evaluating Customer Relationship Management (CRM) activities. Sureshchandra & Shrinivasavadhani (2008) proposed a BSC-based framework for analyzing the strategic impacts of ERP systems (e.g. how the use of ERP system impacts the business objectives of an organization). Eilat *et al.* (2008) proposed a BSC model for the strategic evaluation of R&D projects, in which they deployed the original four perspectives of BSC, each supported by R&D-customized evaluation measures.

2.4 Change Impact Analysis

Change impact analysis was originally introduced in Requirements Engineering (RE) to explore the potential impacts of a requirement change, and estimate the needs of applying the change (Martin & Raffo, 2000). Inspired by the RE impact analysis techniques, a number of approaches were then proposed analyzing impacts of a change in business and software processes (Chou & Chen, 2000; Mujeeb-u-Rehman *et al.*, 2005; Soffer, 2005; Seunghun *et al.*, 2009).

Mujeeb-u-Rehman *et al.* (2005) proposed an algorithm for locating the impact of a change in a **business process**. The proposed algorithm works through propagating the impacts of a process change, over the elements of the business process (actions, objects, and participant), specified in particular process modeling language (KAT). Soffer (2005) investigated the impact analysis of changes in **business process**, and introduced the concept of "scope of a change". The objective of the research was to identify the potential impacts of a change point

on the preconditions, inputs, and outputs of other process activities. The proposed IA solution was explained for process models described in the Generic Process Model (GPM) ontology.

Chou & Chen (2000) introduced a process engineering framework for definition and evolution of concurrent **software processes**. As part of this framework, they implemented a mechanism for defining change plans, and analyzing the impacts of a process change on the rest of process elements. Seunghun *et al.* (2009) proposed an approach for analyzing the impacts of **software process** change, using process slicing. The idea of process slicing was inspired by program slicing techniques, which identify pieces of a code that strongly dependent on each other. In process slicing, first the dependency model of process elements would be identified, and then impact of change in a particular slice would be analyzed.

Most of the techniques that have been so-far proposed for process impact analysis have two common characteristics: first, they are mainly proposed for processes with programmatic specifications, i.e. process models which are represented in a particular process definition language; second, current impact analysis techniques are mainly focused on identifying the impacts within the domain of process, i.e. identifying which other elements of a process would be impacted in case of changing the process. None of the proposed IA approaches investigates the impacts of changing a process over the strategic objectives of the process and organization.

2.5 Current Approaches of Identifying Process Concerns

2.5.1 Process Assessment

Classical Software Process Improvement frameworks (e.g. CMMI, bootstrap, and SPICE) identify the process problems through the *assessment* of as-is software development process. In such frameworks, the assessment procedures are aimed at identifying the *maturity level* of software organization. Therefore, they define different classes of organizational maturity, and then propose guidelines for assessing the maturity level. For example CMMI defines five maturity levels (Initial, Repeatable, Defined, Managed, and Optimizing) for development processes of software organizations. The assessment proceeds by focusing on a sample set of

organization's projects, and through *questionnaires* and *interviews* the maturity level of software organization will be assessed (SEI, 2009).

Process assessment is not the only way of evaluating an organization, as Personal Software Process (PSP) argues that evaluating the capability of individuals (process performers) is as important as assessing the maturity of process for an SPI initiative (Humphrey, 1989). The ISO 1504 is the most comprehensive process assessment model. One of the basic arguments of maturity assessment approaches is that organizations with lower maturity level would not benefit from the advantages of higher maturity levels. In this regard, a set of improvement initiatives will be proposed in order to improve the maturity level of organization.

2.5.2 Process Concern Identification in Small- to Medium-Size Organizations

Maturity assessment methods benefit from their structured and systematic approach, however, they are barely affordable for small- to medium-size companies as they typically need a great deal of time and budget (Derniame *et al.*, 1999). The alternative approach that many companies try for identification of their process-related concerns is the use of light-weight process modeling techniques, along with group- or individual-interviews (Ahonen *et al.*, 2002); (Savolainen *et al.*, 2007; O'Donnell & Richardson, 2008). In these approaches, simple models of the software development process, such as activity- or data-flow models will be developed. The main attribute of these approaches is the emphasis on the participation of everyone in development of process models, and recognition of process problems. In such meetings, each participant will be asked to express his/her idea about the current process, its positive and negative aspects, perceived problems, and potential improvement initiatives. The main advantages of such participatory approaches are the reduction of need for the external process consultants, and in-placement of process improvement as an inexpensive continuous initiative.

Chapter 3. A Framework for Pre-Adoption Analysis of Agile Practices

This thesis introduces a framework for analyzing a set of Agile practices, prior to their enactment in an organization. The proposed framework is called *Strategic pre-Adoption analysis Framework (SAAF)*. This chapter, first, introduces three key components of the SAAF: (1) *Organization Strategic Model*, which represents the key strategic objectives of the organization; (2) *Evidence-Based Repository of Agile Practices*, which contains evidential information (knowledge extracted from empirical studies) of various Agile practices; and (3) the *SAAF Process*, which connects various activities and artifacts of the framework. The overall process of SAAF is divided into three main phases: (A) *Initialization*, which is targeted at setting up the transition team of transitioning to Agile, and constructing the strategic model of organization; (B) *Strategic Agile Practices Analysis*, which is aimed at identifying strategic impacts of enacting candidate Agile practices within the development environment of the organization; and (C.) *Strategic Actor Analysis*, which is intended to clarify problems of the current process of the organization, and investigate whether they can be properly addressed by the selected set of Agile practices. Figure 8 shows the key components of SAAF.

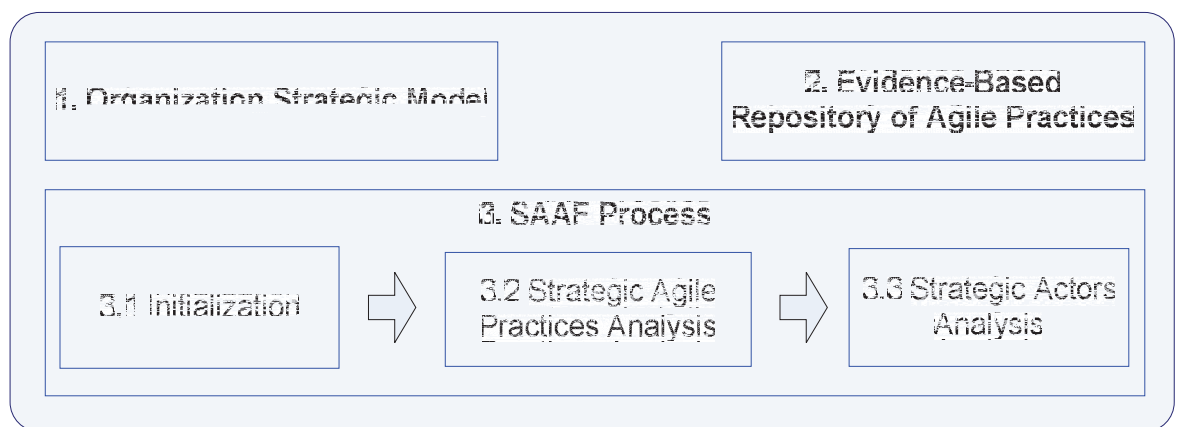


FIGURE 8: BUILDING BLOCKS OF THE SAAF

3.1 SAAF Rationale

As alluded before, SAAF is intended for pre-adoption analysis of Agile practices, which build up an Agile method. The framework cannot be considered as a fully-fledged Agile adoption framework, as it does not introduce any explicit guideline on how to prepare an organization for adopting Agile, or how to customize an Agile method to be adaptable with a particular organization environment. Indeed, SAAF does not provide any guideline on post-adoption stages of transitioning to Agile, which are aimed at the correct enactment of the selected method.

SAAF can be considered as a method evaluation framework, which evaluates a candidate (to-be) Agile method from the strategic viewpoint of an organization. To this end, it takes a model-driven approach for building the organization strategic model, and also evaluating different method proposals. SAAF also aims at facilitating the maximum participation of various organization parties throughout different stages of the framework. In order to improve the reliability of its analysis results, SAAF uses an evidence-based repository of Agile practices, which supplies the results of empirical studies of Agile methods, to the evaluation activities of the framework.

3.2 Organization Strategic Model

In order to analyze a candidate Agile method from the strategic perspective of an organization, the first step is to clarify the key strategic objectives of the target organization, and identify their relationships. SAAF takes a model-driven approach for the extraction and the definition of the strategic models. At its earlier stages, it constructs the Strategies Graph, which captures the strategic goals of an organization. This graph is built following a simplified variant of the *i** Goal- and Agent-Oriented Modeling Framework (Yu & Mylopoulos, 1994). The Strategies Graph is then used in other stages of the framework for subsequent analytical procedures.

The Figure 9 shows a portion of the strategies graph built in our Ericsson project. Further details, including the process of the strategies graphs are presented in Chapter 5.

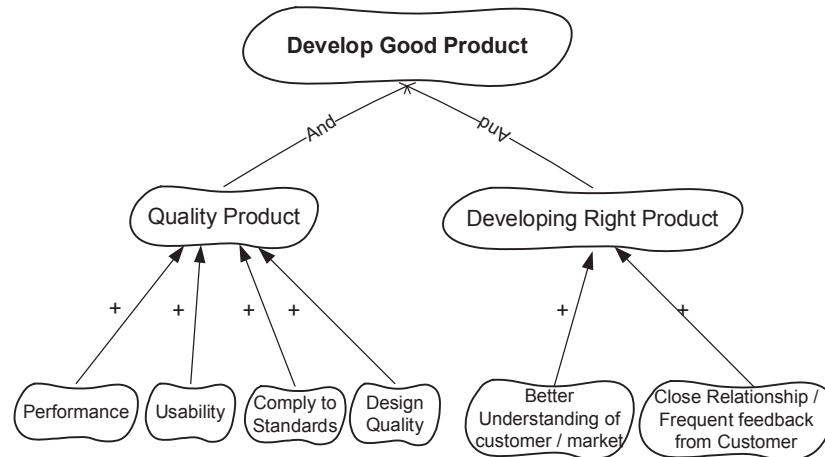


FIGURE 9: A PORTION OF A SAMPLE STRATEGIES GRAPH

3.3 Evidence Based Repository of Agile Practices

One of the key components of the SAAF is its repository of Agile practices. This repository provides strategic information for major Agile practices. The information includes the strategic objectives that can be either positively or negatively impacted by the introduction of a new Agile practice into an organization. Indeed, the repository provides some information about the requisites, which form the pre-conditions of adopting an Agile practice, successfully. For instance, the repository provides the list of over 25 objectives (e.g. “Time to Market”, “Improved Communication”, etc.) which can be impacted by the introduction of the Agile practice “Pair Programming” into an organization. This repository is available online at www.processExperience.org.

The information presented in this repository is evidence-based, as it describes under what circumstances a particular contribution (from an Agile practice to a strategic objective) was observed. The information gathered in this repository has been extracted out of a Systematic Literature Review of published empirical studies on Agile methods. Our industrial experience at the Ericsson Company showed that process managers highly appreciate the content of this

repository, as it helps then getting a huge amount of information about various Agile practices in a very short time.

The repository by itself is one of the major contributions of this thesis to the software engineering body of knowledge. The repository supports the subsequent strategic analyses, which will be described in the other component of framework – Strategic Agile Practices Analysis. Details of this repository, including the research method of building the repository, are described in Chapter 4.

3.4 SAAF Process

3.4.1 Initialization Phase

Step 1.1 – The first step of the initialization stage is to form the transition team. An ideal transition team should be composed of the representatives of various organization roles. The transition team in SAAF is involved in most activities of the framework. It is commonly accepted that constructive participation of various organization parties is one of the key success factors of process improvement initiatives. Therefore, in order to run the SAAF successfully, it is needed to invite right members of organization to the transition team; and keep the involved and motivated until the final steps of the framework.

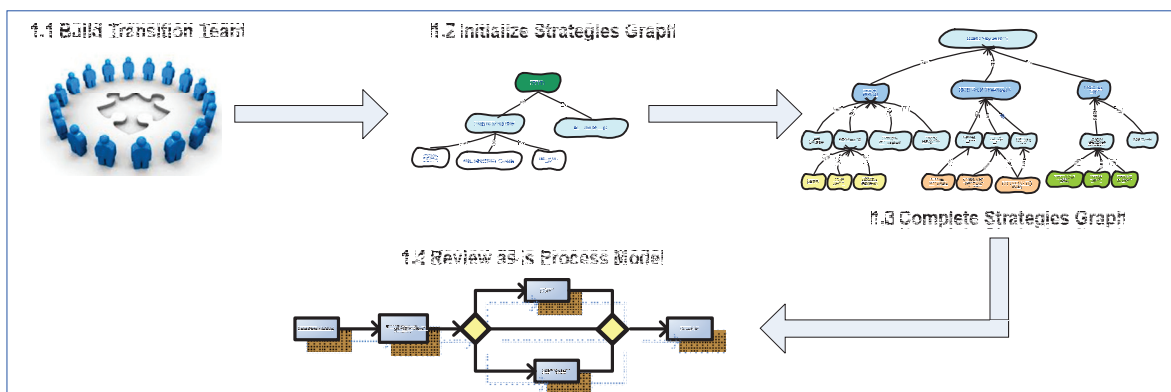


FIGURE 10: PHASE ONE: INITIALIZATION

Step 1.2 – once the focus is formed, or even during the process of its arrangement, the chief members of process analysis team start to build the initial version of the organizational Strategies Graph (SG). These members are often the process manager of organization (or the person who is mainly in charge of transitioning to Agile), together with few selected members of organizational managers, at different levels. The process of building an initial version of the SG should not take too much time, as a one hour meeting should be adequate. The initial SG is containing only the high-level strategic objectives of organization, without delving into their details.

Step 1.3 – the initial version of the Strategies Graph would go through a number of iterations, and by participation of all members of transition team will be completed. Initial (or semi-completed) version of SG will be distributed to selected sub-teams of the transition team, asking them express any objective, which might strategically matter to the organization. The collective wisdom plays an important role in the completion of strategies graph, as it is intended to reflect the key strategic concerns of all parties within an organization. This approach of constructing strategic models has been studied to result in not only more comprehensive strategic models, but to help various organization members having a shared and unified understanding of what matters to the organization.

Step 1.4 – the final step of initialization phase is to review the as-is process of software development, by first, key members of process analysis team, and second members of the transition team. The intention behind this step is to make sure that members of process transition team (transition team members and process analysis team) have a similar understanding of the current process model. The review of as-is process can be facilitated if the graphical model of current process be available (in standard process modeling languages such as UML Activity Diagrams, or Business Process Modeling Language). However, the existence of such models is not mandatory, and any other approach (even informal discussions) which serves the purpose of this step would suffice.

3.4.2 Strategic Agile Practices Analysis

The second phase of framework is about analyzing candidate Agile practices, with respect to the strategic objectives of the organization. The primary prerequisite of this stage is the Strategies Graph of organization, developed at the previous section. The set strategic analysis

procedures proposed at this phase of framework help process analysts to gain a clear understanding of the strategic effects of the candidate new Agile method, in the target organization. This phase of the framework takes the advantage of the Evidence Based Repository of Agile practices.

Step 2.1 (Strategic Contribution Analysis) – explores *direct* contributions of any candidate Agile practice to the strategic objectives of organization. The outcomes of this step are, first, the list of strategic objectives, which can be contributed by any Agile practice, second, the type of contributions (positive or negative) made to any objective, and third, the rationale behind contribution relation. For instance, the Strategic Contribution Analysis of the Agile practice Pair Programming, might result in identifying 10 strategic objectives of the organization to be directly contributed by PP, among which would be “Shorter Time to Market (STM)” and “Lower Defects in Code (LDC)”. The analysis will clarify that the STM will receive a *negative* type of contribution from the PP, as more resources should be allocated to same task; but the LDC will receive a *positive* contribution type from PP, since programming often results in better quality code. Such kind of information about Agile practices is being supplied to this step of framework from the Evidence Based Repository of Agile Practices.



FIGURE 11: PHASE TWO: STRATEGIC ANALYSIS OF CANDIDATE AGILE PRACTICES

Step 2.2 (Propagative Strategic Analysis) – investigates the *indirect* contributions of the candidate Agile practices to the organization strategic objectives. Following the PP strategic analysis example, the strategic objective LDC will receive positive direct contribution from the strategic analysis of PP in almost all organizations. However, in a software company which

builds safety critical systems, the LDC might be under a high-level strategic objective “Saving People’s Life”, whereas in some other organizations it might just be under “Saving Organizational Reputation”. The Propagative Strategic Analysis would clarify the kind of contributions which can be made to such high level objectives of the organization. Indirect contributions are identified based on an algorithm for propagating the direct contribution relations. This algorithm is inspired by an evaluation technique, originally proposed for the core *i** modeling framework.

Step 2.3 (Aggregative Strategic Analysis) – clarifies the overall strategic impact an candidate method, rather Agile practices, on the strategic objectives. The two previous steps of the strategic analysis were evaluating candidate practices independently, however, Agile practice which build up a new Agile method often complement each other, and in order to understand the overall strategic impacts of a candidate Agile method, it is required to aggregated the analysis results of its practices. The result of Aggregative Strategic Analysis is quite helpful in comparing alternative Agile methods, which differ in their practices.

Step 2.4 (Strategic Balance Analysis) – examines whether the impacts of new Agile method is making a balanced across different categories of organizational objectives. This analysis is mainly intended to prevent the adoption of Agile practices, which improve a particular set of organizational objectives, at the cost of devastating hurt to some other objectives.

3.4.3 Strategic Actor Analysis

For many software organizations the need for transitioning to a new process would become apparent by the emergence of inefficiency symptoms at the current development process. Being late to market, unhappy developers, ever-complaining customers, and over the budget projects are some of the common concerns of software organizations, which plan for any improvement in their development style. SAAF recognizes that any process improvement initiative – either in the form of introducing a completely new process model, or just changing some aspects of the current process – cannot be successful, unless the organization gains a clear understanding of its as-is process concerns, and selects a to-be process model which targets the removal of those concerns.

The third phase of SAAF is focused on the extraction and the analysis of process problems; though, most of its activities can be done in parallel with the second phase. The removal of these problems, which are referred *as as-is process concerns*, are in most cases the main reason of software companies for changing their as-is process, and transitioning to a new process model, e.g. Agile. This phase of the framework is inspired by the classical Software Process Improvement frameworks. Most certainly, having a right perception of what are going wrong, what are the reasons behind them, and what are the actual consequences these problems can guide process analysis team to come up with selecting a process model, which has the potential of removing a higher number of these issues. Figure 12 displays an overview phase two activities and artifacts.

Step 3.1 – the starting point of phase three is to conduct a series of interviews with members of the transition team. These interviews are intended to collect the as-is process of concerns of organization, from the viewpoints of its different parties. Members of transition team can be interviewed individually or in groups of two to three. Detailed guidelines of such interviews are presented in chapter 6 of the thesis. The interview sessions will be continues along the second and third steps of this phase.

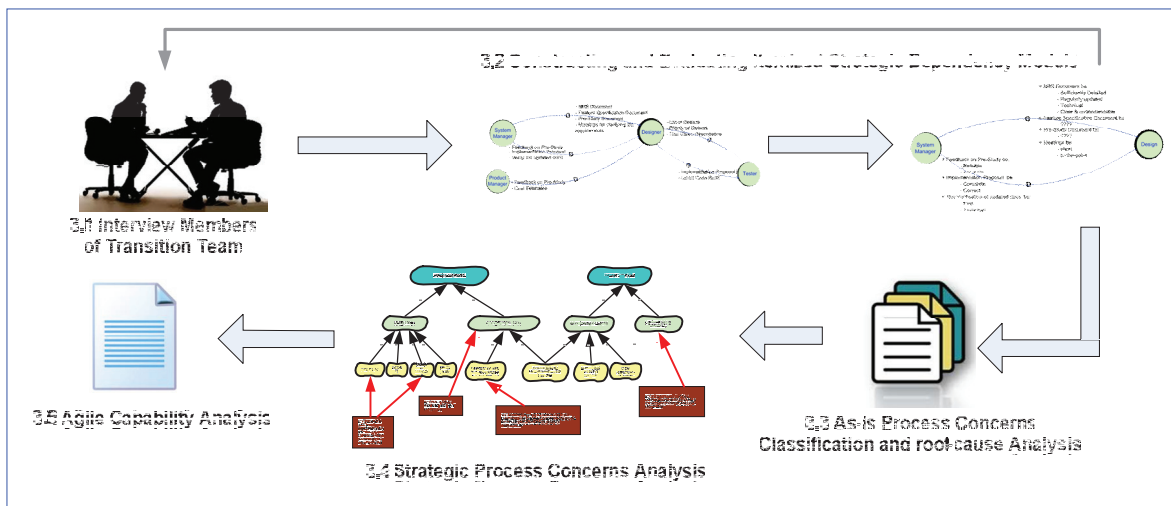


FIGURE 12: PHASE THREE: STRATEGIC ACTOR ANALYSIS: IDENTIFYING AS-IS PROCESS CONCERNS

Step 3.2 – SAAF introduces a modeling approach for the extraction of as-is process concerns. The proposed modeling approach is being used throughout the interview sessions to help interviewees identify the process concerns, affecting their daily work. The proposed modeling approach, called *Itemized Strategic Dependency Modeling*, is inspired by the *i** Strategic Dependency diagrams, and seeks the extraction of process concerns by highlighting dependency relations, which exist among various organization members. It then works by focusing on what *functionalities* or *quality attributes* have been missing in the mutual dependency relations. The application of this technique in Ericsson project resulted in expression of over 100 issues, which were concerning different organization members.

Step 3.3 – The next step of this phase is to harmonize the terminology, categorize the as-is concerns, and conduct a root-cause analysis among them. Different members of an organization might use different terminologies for referring to one process concern. Besides, interviewees often address process concerns at different levels of granularity and importance, all together. The result of this step is often perceived by organization managers as a valuable source of information on what are the current problems of the organization, and why they have been emerged. This step is similar to the *Process Assessment* phase of classical Process Improvement frameworks.

Step 3.4 – the next step of this phase is to investigate the impacts of as-is process concerns on the strategic objectives of the organization. SAAF defines a specific procedure for clarifying what strategic objectives of the organization are being negatively affected by any of the as-is process concerns. This analysis is crucial for making a strategic transition to Agile, since it clarifies the strategic objectives, which have been hurt, and should be supported by new process model.

Step 3.5 – investigates capabilities of candidate Agile practices in addressing as-is process concerns. This step has a significant role on the approval or rejection of a candidate process model, as if it fails to address key concerns of the organization, it would not worth to be invested upon.

Chapter 4. Evidence Based Repository of Agile Practices

Given the growing number of published empirical studies about using Agile in different project situations, it is now possible to gain a more realistic view of what each Agile practice can accomplish, and the necessary requisites for its successful deployment. This body of knowledge can greatly impact the process of adopting a new Agile method, by providing empirically tested evidences about the success and failure scenarios of Agile methods. With the aim of making this knowledge more accessible, this section introduces a repository of Agile practices, which organizes the evidential knowledge according to their objectives and requisites. The knowledge is gathered through systematic review of empirical studies which investigated the enactment of Agile methods in various project situations. In addition, a modeling paradigm is proposed for visualizing the stored knowledge of Agile practices, to facilitate the subsequent steps of the SAAF.

In recent years, Agile methods for software development have seen widespread adoption and experimentation. Agile methods are appealing partly due to their inherent characteristics such as simplicity, and partly due to problems with traditional methods (Boehm & Turner, 2003; Cohn & Ford, 2003). The prevalence of Agile methods has provided a suitable test-bed for software researchers, to run empirical studies, and analyze different aspects of Agile methods in real trials. Every year, a considerable number of empirical studies emerge, each addressing specific concerns of deploying Agile methods in various projects/organizations. These studies, which mostly take macro-process view to the research of Agile processes (Osterweil, 2005), form a large body of knowledge about various fragments (constituent pieces) of Agile methods. For instance, a simple title-based search in just the IEEE Explorer for "Pair Programming" returns over 50 results, mostly reporting on empirical studies.

Despite the abundance of empirical data for Agile methods, there is no systematic support to take advantages of that knowledge, to make it more usable and accessible. This issue becomes more serious when we consider the way that software industry deploys Agile methods. For many project managers Agile software development means to deploy a number of Agile practices, and then try as hard as possible to get the works done. Based on a recent survey, around 60% of software companies do not follow any particular method, and just deploy some techniques and tools introduced by different methods (Bygstad *et al.*, 2008). This approach exposes a software company to the risk of adopting inappropriate development methods, which is a serious threat to the success of software projects (Slaughter *et al.*, 2006). Most likely, the complicated solutions of process (or method) engineering would not be acknowledged by many Agile software companies. But, a promising solution can be a one that facilitates their access to the experience of other practitioners in using different Agile practices, and helps them decide on the appropriateness of their own process.

This chapter introduces a structured repository of Agile practices. For each Agile Practice (AP), the repository entry states the objectives that the AP aims to contribute to, and a set of requisites that are needed for its success. The knowledge of this repository has been gathered through systematic review (Kitchenham, 2004) of empirical studies on Agile methods, i.e., the objectives and requisites of each AP have been synthesized by reviewing results from experiments or reported experiences, not from the motivational description of an Agile method. The repository is evidence-based, as it provides contextual evidences from empirical studies, which explain situations in which a particular objective or requisite of an AP had been met or not. In addition, this repository offers a visualization of AP data, using a goal-oriented notation.

Taking the evidence-based approach for populating this repository helped us to present some interesting results about the contextual aspects of APs. There exist numerous reports that studied the impact of different project contexts on the success or failure of APs. For instance, this repository contains project contexts in which Pair Programming (as a sample AP) either helped or impeded a project to be on-time to market. The evidential knowledge of this repository can be useful for project managers, in order to come up with more appropriate sets of Agile practices for their particular projects/organizations. It can also facilitate the introduction of a new Agile method to an organization, and the reuse of its prior successful and

unsuccessful experiences. The repository is being tested at a major telecom company in a software process improvement initiative.

4.1 Current Repositories of Agile practices

A number of process engineering frameworks include method repositories as components. For instance, the *Open Process Framework* (OPF) (Firesmith) is a process (method) engineering framework, which contains a repository of Agile practices. Similarly, the *Eclipse Process Framework* (EPF) (EPF, 2006) is an Eclipse-based process engineering tool, which also provides a number of libraries for some well-known methodologies, e.g., RUP and Scrum. These repositories are mainly aimed at describing the constituting elements of a methodology by breaking it down to a number of fragments, and then describing each Agile practice in terms of its type, functionality, needed skills, relevant roles, etc. However, these repositories are not concerned about presenting actual evidences about the success or failure of various Agile practices under different project situations.

4.2 Evidence-Based Software Engineering

Evidence-Based Software Engineering (EBSE) has been introduced to take advantage of practitioners' experience with using different tools and techniques of software engineering (Kitchenham *et al.*, 2004). One of basic approaches of EBSE is *Systematic Review*, whose goal is to gather results from different empirical studies, and through analysis and synthesis of extracted data, draw reasonable conclusions (Kitchenham, 2004). Given the growing number of empirical studies about Agile software development methods, systematic review can be a proper approach for building a repository of Agile practices including pragmatic evidences from experiences and experiments. This approach has been used for Agile methods, e.g., in the *Software Engineering Evidence Database* (SEED) (Janzen, 2009), although the presented knowledge was not systematically analyzed.

4.3 Building the Evidence-Based Repository of Agile Practices

The repository is build through the systematic review (Kitchenham, 2004) of published literature on empirical studies of Agile methods. First, to clarify the objective of systematic review, we specified our research questions. Then, we set up a review protocol that specified the criteria of study selection, the strategy of data extraction, and the approach of analyzing and aggregating the extracted data, and synthesizing the results. The final stage of systematic review is the reporting of results, which we publish in textual table format and as graphical models.

4.3.1 Questions of the Systematic Review

Systematic reviews should be targeted towards clear research questions (Kitchenham, 2004; Kitchenham *et al.*, 2004). In this research we were looking for answers to the following questions:

1. What are the actual beneficial outcomes achieved by an Agile practice? These will be treated as *objectives* in the repository entry so that fragments can be retrieved according to desired outcomes. The description of an Agile practice typically comes with promises about its benefits, in terms of quality goals that it contributes to. For instance, "enhanced quality of communication" is one of the major promises of many Agile practices. But, is that really the case? Is there any evidence for this claim? Do Agile practices always contribute positively to their objectives, or can they produce adverse contributions under specific situations?
2. What unanticipated side-effects have been discovered when the Agile practice was put to use under different conditions? These issues will be treated as objectives to which an Agile practice negatively contributes. For instance, Pair Programming (PP) can result in the reduction of Lines-of-Code (LOC) per programmer per month. Thus, PP makes a negative contribution to the productivity of individuals (with respect to the LOC factor). In this research we are looking to identify such side-effects, as well as evidences for their appearance in empirical studies.
3. What are the necessary conditions for the Agile practice to be used successfully? These will be treated as requisites in the repository entry of each AP. The description of every Agile practice typically suggests a number of requisites for its successful enactment. However, these requisites might not be attainable or necessary in all project situations. For instance, "equal engagement of pair in

programming" is one of the requisites of "Pair Programming". But, is it always attainable? In what situations is this requisite most likely to be denied? Furthermore, empirical experience with an AP may have revealed additional requisites.

Here, we stress the importance of finding actual evidences for the claimed objectives and requisites of APs. Such evidences will improve the reliability of the repository, provided that they correctly represent the situation of source evidences. Thus, we had to carefully consider those particular aspects of the reviewed empirical studies (project/organizational factors), which could impact the enactment of their APs. Following this approach, the final repository will contain a considerable amount of situational experiences of APs, which can be used in process improvement initiatives.

4.3.2 Study Selection

We started the research by selecting 20 Agile practices for consideration. For each practice we looked up published empirical studies that explained the enactment of that practice in a particular project/organization. We searched the major collections of computer science publications, including IEEE Explorer, ScienceDirect, SpringerLink, and ACM Digital Library. We used search queries that contained the name of the practice, the name of the originating Agile method, and some phrases for retrieving empirical studies, such as: Case study, Experiment Report, Ethnography, or Survey.

In addition to published empirical studies in academic journals and conferences, we also considered articles from Agile web forums as complementary sources for our reviewed empirical studies. Every selected paper or article was then evaluated with respect to the following items: 1- reliability of reported data; 2- reproducibility of study; 3- relevance to the research questions; 4- quality of situation description (context for enacting AP). Only publications that meet all of the above criteria are used as sources, and are listed as references.

4.3.3 Data Extraction

While reviewing the selected empirical studies, we were looking for either explicit or implicit answers to our research questions. For data extraction, we obtained answers for the following questions for every reviewed paper:

- What is the type of this empirical study? (Controlled Experiment, Survey, Interview, Experience report, Ethnography, Case Study)
- What Agile practices have been investigated in this study?
- What are the distinguishing characteristics of the context of this study? (Human/organization factors, such as cultural issues, structure of development teams, and the distribution of expertise; project related factors, such as complexity and market pressure)
- Does the paper describe the benefits/side-effects of deploying an Agile practice in terms of impacting some method objectives? If yes, what are those objectives? How are they impacted, i.e., did the Agile practice help to attain those objectives, or had the opposite effect?
- Does the paper highlight any situational factor that particularly impacted the enactment of Agile practice? If yes, what are those factors? How do they influence the enactment of practice (improving or impeding)?

We did not begin the systematic review with a preconceived list of objectives or requisites, since one of the purposes of the research was to identify the evidential objectives and requisites of APs. However, after reviewing and analyzing a number of publications for each AP, such lists began to emerge. Afterwards, we were considering those lists while reviewing a paper, and also reread some of the initially reviewed papers.

4.3.4 Data Analysis and Aggregation

The extracted data from reviewed empirical studies were then analyzed and aggregated as units of knowledge for every Agile practice. Data analysis was performed in order to:

- Clarify the contributions of Agile practices to objectives. The contribution relations were represented qualitatively, using a five-value scale (--, -, ?, +, ++). Negative contributions represent the cases where an Agile practice was reported to strongly deny (--) or somewhat deny (-) an objective; unaffecting contributions (?) represent cases where an study reported that an AP does not have any significant contribution to a

particular objective; and positive ones represent the cases where an Agile practice was reported to strongly satisfy (++) or somewhat satisfy (+) a quality objective.

- Resolve the naming problems that were due to synonyms or homonyms. Empirical studies use different terms and taxonomies to describe process objectives. We analyzed the extracted data to harmonize the various terminologies of reviewed studies. Table 7 shows a subset of a harmonized taxonomy of AP objectives.
- Classify the method objectives and requisites into two-level categories. The extracted objectives (or requisite) were classified either as major or minor categories. Minor categories are considered as subcategories of major ones, and represent more delicate aspects of a major quality objective (or requisite). Although the objectives and requisites of an Agile practice can be detailed with further levels of refinement, for the sake of simplicity, we consider only two levels for the current design of the repository. Table 5 and Table 7 show two examples of this categorization.


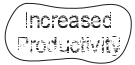


The analyzed data were then aggregated with what had been collected previously. The knowledge of each Agile practice was aggregated in two tables, one for its objectives and the other for its requisites. Data aggregation for an AP was either in the form of adding a new objective or requisite; or providing a new piece of evidence for an existing contribution relation. The final stage of data aggregation was to make generic snapshots, which are two dimensional tables that show the identified (and unidentified) relations of all Agile practices to all objectives or requisites. Generic snapshots facilitate the comparison of Agile practices, and provide insights for further empirical studies to uncover the potential relations between Agile practices and process objectives or requisites. Table 7 shows a sample generic snapshot.

4.3.5 Data Visualization

An appropriate modeling paradigm for visualizing the collected data of this study should be capable of representing method objectives and requisites, also the contribution relations of APs to their objectives and their dependency relations to their requisites. These features have been well-addressed in Goal-Oriented Requirements Engineering (GORE), where goal models are used to express how domain objects (such as activities and resources) contribute to domain goals (Yu, 1997). In this research we use the *i** goal-oriented modeling framework for

visualizing the collected knowledge of APs. Table 4 describes a subset of *i** modeling elements, which are used for visualizing the objectives and requisites of Agile practices. Figure 14 shows an example of AP visualization.

Table 4: The use of *i** modeling notation for visualizing Agile practices

| <i>i*</i> Modeling Element | Description | <i>i*</i> Graphical Notation |
|----------------------------|---|---|
| Task | Represents the enactment of an AP |  |
| Softgoal | Represents an AP objective or a requisite |  |
| Contribution Relation | Visualizes the contribution relation of an AP to its objectives |  |
| Decomposition Relation | Visualizes the dependency relation of an AP to its requisites |  |

4.4 Results

As mentioned before, the results of data extraction and analysis for each Agile practice are aggregated into an objectives table and a requisites table. As an example, this section presents a subset of the knowledge for two APs: "Pair Programming" and "Daily Scrum Meeting". The complete set of tables are publicly accessible at: <http://www.processexperience.org/>.

Figure 13 shows the metamodel of the repository. Each *Agile practice* is contributing to a number of *Objectives*, and requires a set of *Requisites*. Objectives and requisites are categorized into the *Major* and *Minor* groups. Contributions of an Agile practice to its objectives can be of the following types: [++, +, -, --], and every contribution relation is associated with a *Situation*, in which the contribution relation was observed. In description of the situation, the following factors have been considered [Environment, Project, Constraint, Project / Organization]. The dependency of every Agile practice to its requisites is captured by

Decomposition Relation. An *Achievement Status*, and *Situation* description is associated with every decomposition relation, which clarify to what extent (Satisfied, Partly Satisfied, Unknown, Partly Denied, Denied) the subject requisite was achieved in the given situation.

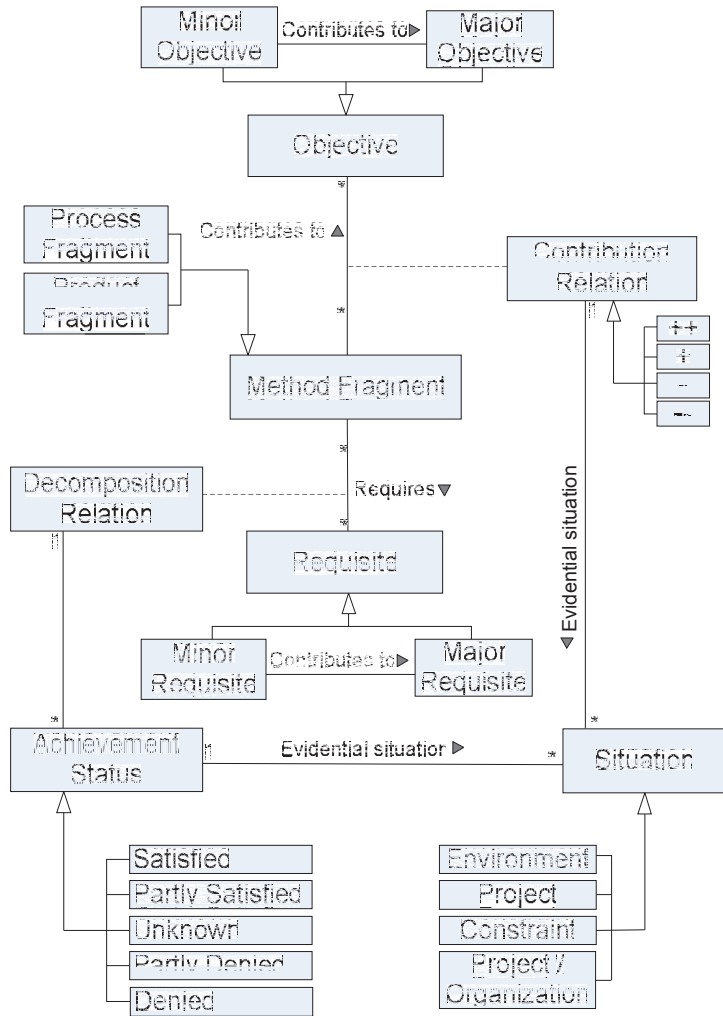


FIGURE 13: METAMODEL OF REPOSITORY

Design Rationale. Why this metamodel is selected for organizing the repository content? As mentioned before, this repository is part of a larger framework (SAAF), and its content are supposed to be used in different stages of the framework. Since the framework takes a goal-oriented approach for the strategic analysis of Agile practices, it requires the repository to provide evidences about goals (or objectives) to which agile practices would contribute. Therefore, the design of repository is based on the requirements of the overall framework, and that had shaped structure of repository and its supporting literature review.

4.4.1 Agile Practice Specific Objectives Dataset

For each Agile practice, the repository provides an *objectives dataset*, which represents the quality goals that are expected to be achieved by the enactment of the Agile practice. These objectives have been extracted from published empirical studies on Agile practices (i.e., the dataset does not include quality goals that were just claimed for an Agile practice without any supporting empirical evidence). This dataset stores method objectives in two categories: *major* and *minor*. A major objective is defined as a quality goal that can be decomposed into a number of sub-goals, called minor objectives. Perhaps the classification of quality goals could be performed more elaborately; however, for the sake of simplicity this framework considers only these two levels. The objectives dataset also provides situational evidences for the contribution of Agile practice to its objectives. Besides, for every contribution relation, the dataset provides the reference to the study that provided the empirical evidence, and possibly the description of the situation of study. Table 5 shows a portion of the objective dataset for Agile practice “Daily Scrum Meeting”.

Typically the contribution of an Agile practice to its objectives is positive. However, there might be some situations where an Agile practice adversely impacts its objectives. For instance, although “Daily Scrum Meetings” usually makes strong positive contribution to the “Improved Awareness” of a team about the activities of other team members, as studied in [S1], in the case of large projects with multiple development teams, daily meetings can cause confusion by bringing up excessive details, which are not relevant for a large portion of developers. The existence of conflicting evidences for the contributions of an Agile practice to one objective, is a valuable knowledge, which further certifies that Agile adoption must be carried out with respect to the situation-specific characteristics of an organization. The framework proposes four possible types of contribution relations: *strongly positive* (++), *positive* (+), *negative* (-), and *strongly negative* (--).

TABLE 5: A SUBSET OF MAJOR AND MINOR OBJECTIVES THAT “DAILY SCRUM MEETING“ CONTRIBUTES TO THEM, WITH REFERENCE TO THE INVESTIGATING EMPIRICAL STUDIES, AND PARTICULAR SITUATIONAL EVIDENCES

| Major Objective | Minor Objective | Contribution Type from Fragment | Study | Situation |
|-------------------------|--|---------------------------------|-----------|--|
| Effective Communication | | ++ | [S1] | Default |
| | Improved awareness (of what others are doing) | ++ | [S1] | Default |
| | | - | [S1] | Large projects, as they may need extensive number of meetings |
| | Real-time knowledge transfer | + | [S8] | Default |
| | | - | [S2, S12] | Distributed Development: use of email and wiki pages for comm. |
| | Enhanced Communication with business people / project leader | ++ | [S3, S8] | Existence of multi-level Scrum in case of many scrum teams |
| | Better understanding of customer needs | + | [S8] | Default |

4.4.2 Agile Practice Specific Requisites Dataset

The other dataset kept for each Agile practice is the requisites dataset. This dataset contains the conditions that should be met for the successful enactment of an Agile practice (e.g., resources to be provided, tasks to be performed, or personnel qualifications to be met). The framework defines the relation of an Agile practice to its requisites as *decomposition relation*, since the successful enactment of an Agile practice is due to the successful achievement of its requisites. Similar to the objectives dataset, this dataset represents method requisites in two levels of abstraction (major and minor), also sets the contribution relation of minor requisites to major ones. Besides the dataset provides situational evidences for each requisite, by referencing to the studies in which the requisite was satisfied or denied (partially or fully). In

most cases it explains the significant situational factors of the referenced empirical studies that affected the fulfillment of method requisites.

Table 6 shows a subset of the requisites of “Pair Programming”, focused on “Effective Collaboration”. For instance, it shows that “Equal engagement (of pairs) in Coding” is a minor requisite that contributes positively to the major requisite “Effective Collaboration”. However, not every situation can satisfy this requisite. For example, the empirical study [S15] has shown that pairing programmers with different levels of expertise can result in passiveness of the weaker programmer, thus partial denial of the requisite. The objectives dataset also takes a goal-oriented approach in representing method requisites in order to facilitate their modeling and evaluation in later stages of the framework.

TABLE 6: A SUBSET OF REQUISITES OF “PAIR PROGRAMMING“, CONTRIBUTION OF MINOR TO MAJOR REQUISITE; SITUATIONAL FULFILLMENT STATUS [SATISFIED(✓), DENIED(×), OR PARTIALLY DENIED(⊗)]

| Major Req. | Minor Requisite | Contrib. to Major Req. | Situa. Fulfill. Status | Study | Situation |
|-----------------------------------|----------------------------|------------------------|------------------------|--|---|
| Effective Collaboration | Equal engagement in coding | + | ✓ | [S15] | Pairing programmers with equal expertise |
| | | + | ⊗ | [S15] | Pairing programmers with different expertise (weaker programmer became passive) |
| | Joint Decision Making | + | ⊗ | [S15] | Similar pairs; the one who had the control of machine usually had a significant advantage w.r.t decision making |
| | Collaboration be viable | + | ✓ | [S17] | Pairs with heterogeneous personality profile |
| | | | ⊗ | [S17] | Pairs with homogenous personality profile |
| Similar working and resting hours | + | × | [S30] | Pairs with different times for starting their job or resting | |

4.4.3 Agile practice Visualization

Figure 14 shows a sample visualization of an Agile practice. It depicts some of the objectives to which pair programming contributes. For instance, it shows that pair programming makes a positive (+) contribution to the objective "Better Time to Market". We used four types of contribution relations (adopted from *i** framework), in order to represent the contribution of an AP to its objectives: "+" representing positive contributions; "++" representing strong positive contributions; "-" representing negative contributions; and "--" representing strong negative contributions. Table 6 also shows the dependency of the AP pair programming on some of its requisites, e.g., "Effective Collaboration of Pairs".

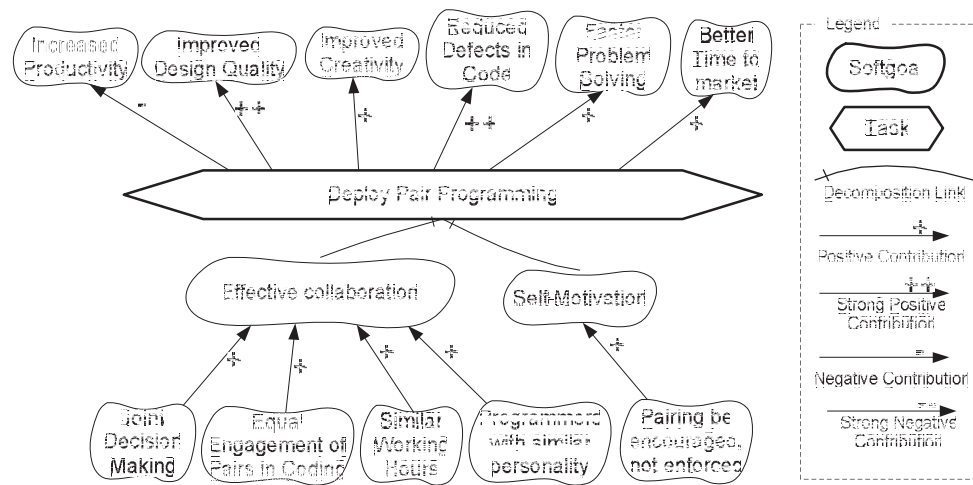


FIGURE 14: VISUALIZATION OF THE AGILE PRACTICE “PAIR PROGRAMMING” FOR A SUBSET OF ITS OBJECTIVES AND PREQUISITES

4.4.4 Agile practices: Generic Snapshots

One of the results of this study is a set of generic snapshots, which summarizes the relations of all (or several) APs to all (or several) of the identified objectives or requisites. These snapshots are called generic because they are not bound to any particular AP, and typically represent a number of APs. For example, Table 7 shows a list of minor objectives of five Agile practices, all related to the major objective "Improved Efficiency". Due to space limits we could not present the contribution of other Agile practices to these objectives, as well as other major objectives. The online version of the repository contains similar tables for other major objectives, e.g., "Improved Communication", "Improved Collaboration", "Higher Job

Satisfaction", and so forth. Note that an empty cell in the table does not necessarily mean that there is no contribution relation between the corresponding Agile practice and objective. Further investigations in the empirical studies of software processes might supply further contribution relations to the table.

TABLE 7: GENERIC SNAPSHOT: A LIST OF MINOR OBJECTIVES OF FIVE AGILE PRACTICES, ALL RELATED TO THE MAJOR OBJECTIVE "IMPROVED EFFICIENCY", AND THE CONTRIBUTIONS OF THE AGILE PRACTICES TO THEM

| Major Objective | Minor Objective | Pair Prog. | Daily Scrum Meeting | Open Office Space | On-site Customer | Time-Boxing + MoSCoW |
|------------------------|--|------------|---------------------|-------------------|------------------|----------------------|
| Improved Effectiveness | Increased productivity (LOC/Month/developer) | - - | | | | |
| | Reduced Development Cost | - , ? | | | | |
| | Improved Design (product) Quality | + + | | | | - |
| | Improved Creativity Potential | + | | | | |
| | Reduced Defects in Code | + +, ? | | | | |
| | Faster Problem Solving | + , ? | | | ++ | |
| | Be on-time to market | + , - | | | | ++ |
| | Elimination of redundant tasks | | | ++ | | |
| | Earlier Detection of Dev. issues | | | ++ | | |
| | Reduced need for documentation | | | | ++ | ++, - - |
| | Be Focused on work | | | | -- | |
| | Balancing conflicts of customer needs and developer's expectations | | | | | ++ |
| | Fast Resolving of requirement issues | | | | | ++ |
| | Controlled Risks per Iteration | | | | | + |

4.4.5 Catalogue of Agile Objectives

Though the initial goal of research was to explore what are the contributable objectives of each Agile practice, after a while, the research resulted in a large collection of objectives, which are somehow contributable by various Agile practices. This collection is represented as a catalogue of Agile objectives, and is available on the repository web portal.

As mentioned before, the data extraction phase of this research was not based on any initial set of objectives, as the research was aimed at identifying such objectives. Therefore the catalogue of Agile objectives has been generated while the research activity was going. The extracted Agile objectives were grouped into the 10 major categories of objectives. Table 8 lists the identified major objectives, also the number of Agile practices that somehow contributed to each objective, and the number of minor objectives that are identified under the category of each major objective. The online version of repository provides a feature that by selecting every major or minor objective, its list of all contributing Agile practices, as well as their situational evidence, will be shown.

TABLE 8: IDENTIFIED MAJOR OBJECTIVES, NUMBER OF CONTRIBUTING AGILE PRACTICES, AND NUMBER OF GROUPED MINOR OBJECTIVES (BASED ON THE REPOSITORY STATUS ON THE MAY 2012)

| Major Objective | #of Contributing APs | #of Minor Objectives |
|--------------------------------------|-----------------------------|-----------------------------|
| <i>Enhanced Project Visibility</i> | 3 | 4 |
| <i>Improved Collaboration</i> | 6 | 13 |
| <i>Improved Communication</i> | 10 | 17 |
| <i>Improved Effectiveness</i> | 17 | 46 |
| <i>Improved Job Satisfaction</i> | 6 | 7 |
| <i>Improved Motivation</i> | 1 | 3 |
| <i>Improved Product/Code Quality</i> | 6 | 8 |
| <i>Improved Project Management</i> | 5 | 9 |
| <i>Improved Project Planning</i> | 6 | 12 |
| <i>Improved Team Management</i> | 7 | 5 |

Similar catalogue was expected to emerge for Agile requisites. But, due to the issues of extracting Agile requisites, also their diversity and strong dependency to the project/organization situation, the classification of minor requisites into major ones was not in a form to be presentable as a catalogue of Agile requisites. The complete list of identified requisites is available online at the repository web portal.

Chapter 5. Strategic Analysis of Agile Practices

Recognizing that Agile adoption is often motivated by strategic concerns such as market competitiveness or responsiveness to customer needs, this chapter introduces a set of techniques for the strategic analysis of Agile practices. These techniques facilitate the upfront analysis of how each candidate AP would contribute to which strategic objectives, and how they might complement or interfere with each other. The right application of these techniques can improve understanding within a team and ease the transition to Agile. The presented techniques are mostly based on the previously explained strategic model of organization, and the evidence-based repository of Agile practices.

The chapter, first, introduces the Organizational Strategic Model, and presents a modeling technique for constructing such models. The proposed modeling approach is based on the i^* modeling framework, and has been customized regarding the specific needs of the SAAF. The chapter will then continue by defining five analysis techniques for evaluating the impacts of transitioning to a new Agile process on the organizational strategic objectives. The four techniques are as follows: (1) Strategic Contribution Analysis, (2) Propagative Strategic Analysis, (3) Aggregative Strategic Analysis, and (4) Strategic Balance Analysis. The proposed techniques are all based on the strategic model of the organization. The exemplar application of these analysis techniques is described in detail in chapter 7, which presents the results of SAAF experience in Ericsson.

5.1 Organization Strategic Model

The strategic model of an organization represents the break-down of key organizational vision and mission statements into actionable objectives (Porter, 1996). These objectives can be further categorized into various groups, and complemented by indicators and target values. For instance, a software organization might set “Developing Quality Product” as one of its missions. To this end, the organization should come up with a set of actionable objectives, such

as “Using Reliable Third Party Packages” or “Capturing Code Defects at the Earliest Time”, and if needed, complement them by indicators, such as “Percentage of Defects Found at Each Stage of Development” and target values.

Such models are widely used in strategic management of different organizations, as the basis of important decision makings. These models can be represented in simple tabular format (as the one shown in Table 9). Although such models would represent the categories and supporting attributes of various organizational objectives, they are quite inefficient in representing the interleaving relationships of strategic objectives, such as multi-level break-down or cross-category contributions. Therefore, frameworks which have been proposed for the strategic management of organizations often come with a modeling paradigm, for visualizing strategic models. These models, not only helps representing the relationships of strategic objectives, they can greatly facilitate the process of building strategic models, by visually assisting the participants to better define strategic objectives and their relations.

In SAAF, building the right strategic model of organization is a key to the success of framework, as most of the subsequent analysis activities are based on this model. The framework introduces a modeling paradigm, which is based on i^* modeling framework, to visualize the strategic objectives and their relationships in a model called *Strategies Graph* (SG). This chapter explains: how to build a SG, and how to use SGs for building comprehensive strategic models.

5.1.1 Strategies Graph

The Strategies Graph (SG) is the visualization of organizational strategies. It contains the structural breakdown of organizational strategies at different levels and categories. The Strategies Graph introduces two types of relationships among strategic objectives: *Decompositional* and *Contributinal*, The decompositional relations represent the logical AND/OR decompositions of high-level objectives to those of lower-levels; and Contributinal relations represent the positive and negative impacts of strategic objectives over each other. The contributinal relations can be plotted across different categories (e.g. business, customer, internal process) and levels (high-level visional strategic to low-level actionable objectives) of strategies graph.

Figure 15 shows a sample SG. It shows how the high-level strategic objective “Developing Good Product” is logically decomposed to the “Developing Quality Product” and “Developing Right Product”. It also depicts the positive contributinal relation of the strategic objective “Better Understanding of Customer / Market” towards “Developing Right Product”.

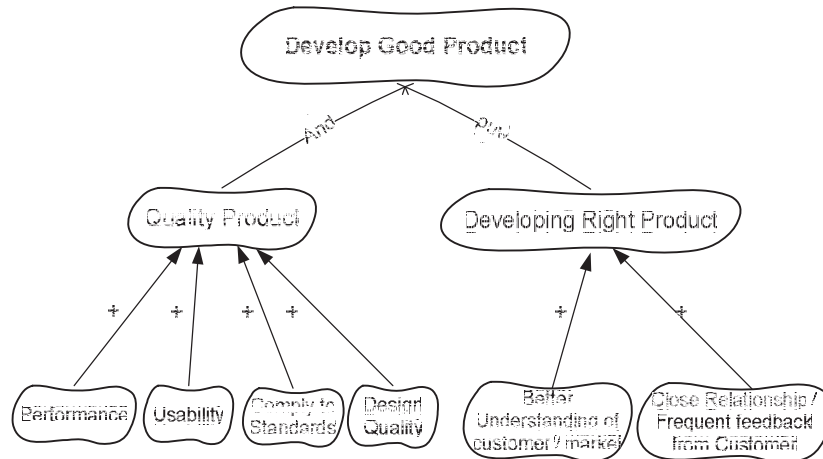


FIGURE 15: A SAMPLE STRATEGIES GRAPH

The Strategies Graph adopts its main constructs from the *i** modeling framework (Yu, 1997). *i** is a goal and agent oriented modeling framework which can be used to represent the strategic aspects of a modeling domain. An evaluation mechanism helps to track the impacts of achievement (or denial) of certain strategic objectives over other ones (Horkoff & Yu, 2009). The *i** concept of *Softgoal* is used to model strategic objectives. The contributinal relations of strategic objectives are represented by a variant of *i** notation of Contribution Link: “++” For Strong Positive, “+” for Positive, “-” for Negative, and “--” for Strong Negative contributions. “AND” and “OR” links are used to represent logical decomposition of strategic objectives.

The Strategies Graph is developed iteratively. The framework stresses the participation of all organizational managers, and suggests the use of a light-weight modeling technique for facilitating the work. A participatory approach is needed to bring various stakeholders’ viewpoints into the model of organization’s strategies. The Strategies Graph also helps participants to achieve shared understanding about organizational strategies, and to uncover differing interpretations or misunderstandings (Kaplan & Norton, 1996b). Similar approaches have been used in knowledge extraction for strategic management (Kaplan & Norton, 2004) and process improvement (Savolainen *et al.*, 2007).

The strategies graph is intentionally designed to use a simple notation, with the minimum number of modeling constructs. The reason behind this design is that the SG is not only used for visualizing the organizational strategic objectives, it is also used as a mean for eliciting strategic objectives out of the tacit knowledge of organizational staff.

5.1.2 Further Elements of Strategic Model

Strategic objectives in the Strategies Graph may be accompanied by optional attributes, such as *quantifiable measures* and *desired target values* (Kaplan & Norton, 1996b). These attributes would be used if an organization plans for monitoring the achievement status of the strategic objectives at various stages of the process adoption.

The construction of the Strategies Graph is assisted by the *categorization* of strategic objectives. The Balanced Scorecards method suggests four perspectives for categorizing organizational strategies: *Financial*, *Customer*, *Internal Business Process*, and *Learning & Growth* (Kaplan & Norton, 1996a). Other classifications have been proposed for strategic objectives of software companies (Martinsons *et al.*, 1999; Grembergen & Amelinckx, 2002). (Martinsons *et al.*, 1999) proposed the following four perspectives on strategies for information systems organizations:

- *Business Value* – Strategies leading to the value of business
- *User Orientation* – Strategies leading to added-value for customer
- *Internal Process* – Strategies leading to efficient development process
- *Future Readiness* – Strategies leading to continuous improvement and preparation for future challenges

The categorization of strategies provides further structure for working with stakeholders during construction of the Strategies Graph. It recognizes the different strategic interests of various parties in an organization, leading to more effective engagement of stakeholders in completing the Strategies Graph, and thus higher confidence and mutual trust in the analysis results. In the Ericsson engagement, the first three categories of objectives were used, as the

R&D unit was involved in a development project in which business, user, and process related objectives constituted the main strategic concerns.

This information is often provided in the *strategic matrix* in conjunction with the strategies graph. Table 9 shows a sample strategy matrix, which specifies strategic objectives, their categories, measures, target values, and associated initiatives.

Table 9: Sample strategy matrix from (Bloomfield, 2002)

| Strategic Theme: Smart, Profitable Expansion | | | | |
|--|---------------------------------------|--|---|--|
| | Objective | Measure | Target | Initiative |
| Financial | Increase % of revenue from new stores | % Revenue from stores opened in last 3 years | > 30% year 1 > 50% year 3 | Marketing to new target markets |
| | | Avg. # of days to breakeven | < 180 days year 1 < 130 days year 3 | Operations review Site selection |
| | Increase sales efficiency | Revenue per FTE | > \$ X year 1 > \$ Y year 3 | Self-service checkout pilot |
| Customer | Acquire new locations | Avg. # daily customers | > X in first 6 mos., > Y in first year, > Z by year 3 | Local marketing/PR campaigns |
| | | # of repeat customers | > X in first 6 mos., > Y in first year, > Z by year 3 | Customer loyalty program |
| Process | Fact-based site selection | Days lag between market selection and site acquisition | < 90 days year 1 < 70 days year 3 | GIS mapping National brokerage contract |
| | Streamline development process | Project duration, site acquisition to opening | < 365 days year 1 < 300 days year 3 | Standardize design/build processes |
| | | % stores open on schedule | > 93% year 1 > 95% year 2 | Web-based project management |
| Learning & Growth | Use business intelligence systems | % eligible employees trained | >90% year 1 >99% year 2 | In-house system training |
| | Integrated knowledge management | # paper forms used | < 200 year 1 < 100 year 2 < 5 year 3 | Corporate digital nervous system |

5.1.3 Validation of Strategies Graph

As the whole framework is based on the strategies graph of organization, it is quite important to make sure that the developed SG is valid. Any mistake in the formation of SG, in terms of missing or wrong strategic objectives can result in the invalidity of framework outcomes. The following attributes are proposed for a valid SG:

- Containing all key strategic objectives of the organization.
- Not containing strategic objectives, which have become obsolete, or they are expected to lose their value after the transition to Agile.
- Placing strategic objectives at the right level of hierarchy.
- Establishing correct relationships among strategic objectives.

In order to minimize the risk of developing invalid strategies graph it is recommended to develop it in an iterative approach, with participation of various organization parties. At the end, managerial confirmation can be considered as a green light for starting the use of SG for subsequent steps of the framework.

5.2 Strategic Contribution Analysis

The foremost step of strategic analysis is to explore contributions of every Candidate Agile Practice (CAP) towards the organizational strategic objectives. The basic idea behind this analysis activity is to figure out What, How and Why any of the strategic objectives of an organization can be affected by each CAP. The analysis activity – like many other steps of SAAF – is carried out in a model-driven approach. In this analysis, every CAP will be exposed to the organizational Strategies Graph, and then the kind of impact (contribution) that it would make on every strategic objectives, as well as the rationale of that contribution will be investigated.

Figure 16 shows an example of exposing a CAP to an SG, for the purpose of strategic contribution analysis. This model was developed as part of our experiment in Ericsson, and visualizes the exposure of CAP “Sprint Planning” to a portion of the organizational SG. The contributions of a CAP to SG elements are color coded (Green for positive, and Red for negative contribution links) in order to distinguish them from the contribution relations which

exist among SG elements. As depicted in this figure, every contribution relation from the CAP to SG has two elements:

1. **Contribution Type** – For specifying *how* the CAP affects an objective. The framework, inspired by the *i** modeling framework, defines four types of contributions: *Strongly Positive* (++), *Positive* (+), *Negative* (-), and *Strongly Negative* (--), where in positive contributions the enactment of CAP would help the achievement of objective, and vice versa for negative ones.
2. **Contribution Rationale** – For specifying *why* the CAP affects the objective. For example, when a CAP like “Sprint Planning” is identified to be making Positive (+) contribution to the objective “Learning”, its rationale is that “the CAP would help new members to learn planning concepts by participating in planning meetings and seeing how experts decide on project tasks”. . The rationale for each contribution relation is visualized as a dashed box, linked to the relevant contribution link through a dashed line.

After the completion of strategic contribution analysis, it is apparent what strategic objectives will be contributed by each CAP. For instance, Figure 16 shows that objectives: “Better Visibility, Learning, Self-Organization, and Empowerment” are about to be positively contributed by the CAP “Sprint Planning”; and the objective “Improved Project Planning” will receive a conflicting (both positive and negative) contribution from the CAP.

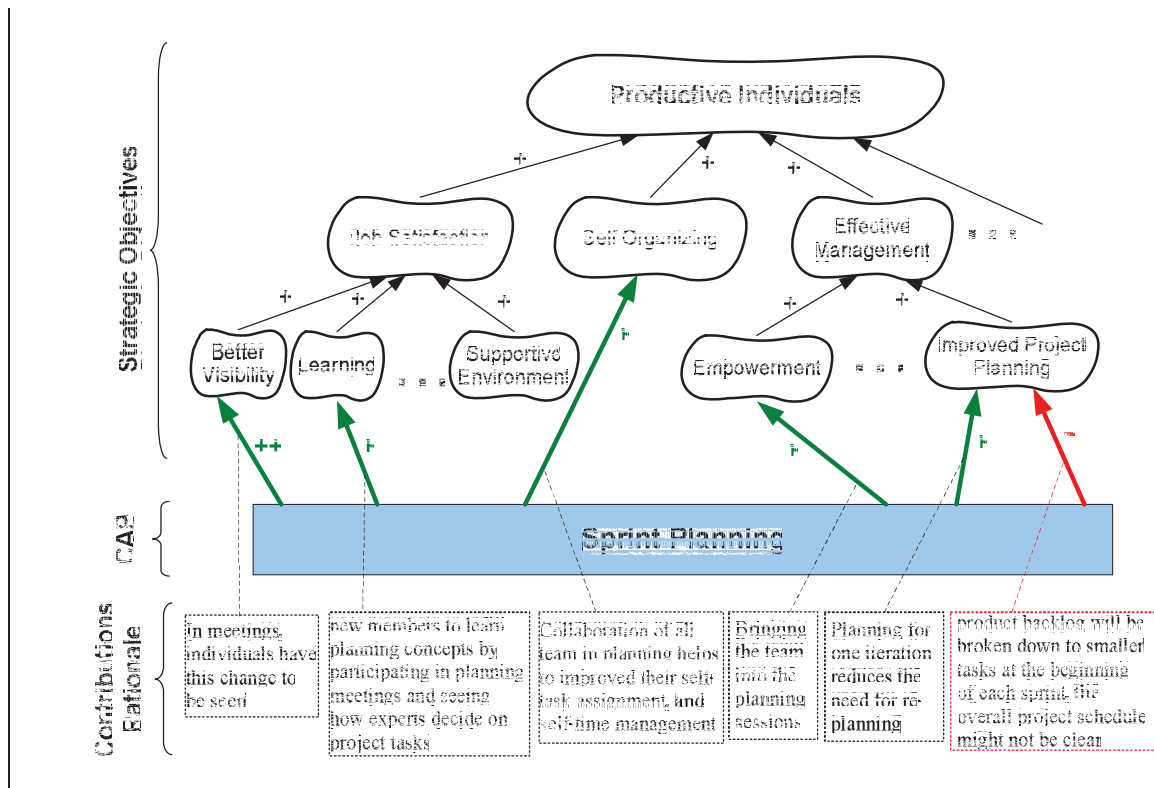


FIGURE 16: (STEP 3) VISUALIZING THE TYPE AND RATIONALE OF THE CONTRIBUTIONS OF THE CANDIDATE AGILE PRACTICE (CAP) "SPRINT PLANNING" OVER A PORTION OF THE STRATEGIES GRAPH (EXAMPLE FROM THE ERICSSON'S CASE)

5.2.1 How to conduct the Strategic Contribution Analysis

The strategic contribution analysis is intended to be carried out in a participatory approach. The recommended approach for conducting this analysis is that in a number of group meetings, which are participated by representatives of various organizational roles, the SG be projected, and for every CAP, the team go through any of the strategic objectives of the SG, and decide whether it can be affected by the CAP. This analysis is quite intuitive, but what supports its correctness and reliability, is the *knowledge base of Agile practices*, which helps the analysis team to identify the *type* and *rationale* of contributions of a CAP to organizational strategic objectives. In fact, SAAF framework proposes two approaches for strategic contribution analysis:

- 1. Evidence Based** – Upon specifying the contribution relation of a CAP over an strategic objective (SO), if the SO be among the built-in strategic objectives of the CAP, or in any way the knowledge base provides some evidence(s) about the

potential contributions of the CAP to the SO, then the analysis team take the advantage of such evidences in order to set the contribution relation of CAP to SO.

2. **Consensus Based** – When the knowledge base provides no supporting evidence, or the provided evidence is perceived to be inapplicable with respect to the organizational situation, the analysis team would decide on the contribution of CAP to SO based on the common agreement of, with respect to the original definition of the CAP and the organizational definition of the SO.

In specifying the type of a contribution relation, the analysis team should consider the possibility of *situational behaviors*. It is possible that a CAP, in some particular situations, impacts an objective differently from its general behavior. For example, the contribution of the CAP “Pair Programming” towards the objective “Be On-time to Market” is situational, in that in some cases the CAP would help, and in some other cases in would hurt the objective. This information is retrieved from the Knowledge Base of SAAF. In this example, the knowledge base states that “when the market pressure is not high, and there is adequate number of developers, pairing programmers would help the project to be on time for market, whereas in other cases it hurts.” Knowing the situational behaviors of a CAP towards an objective allows the analysis team to choose contribution values that are best matched with their own organization and project context.

Figure 17 shows the metamodel of the contributions of a CAP over the strategies graph. The model uses the concept UML association class, in order to represent the elements of contribution relation (from a CAP to an objective); indeed the model represents the categorization of strategic objectives, as well as the relationships which exist among them (Decompositional and Contributinal).

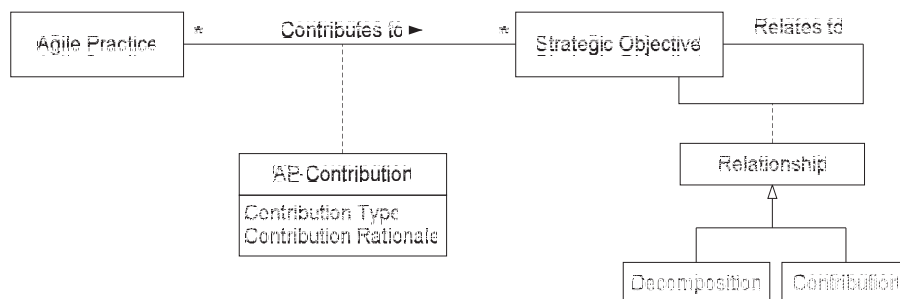


FIGURE 17: METAMODEL OF THE CONTRIBUTION OF AGILE PRACTICES OVER STRATEGIC OBJECTIVES, AND THE STRUCTURE OF THE SG

5.3 Propagative Strategic Analysis

The purpose of propagative strategic analysis is to trace the affects of a candidate Agile practice on a lower-level strategic objective, towards the higher-level ones. In the previous step of the framework (Strategic Contribution Analysis) the analysis team identifies the set of strategic objectives that are *directly affected* by each CAP. This step (Propagative Strategic Analysis) takes as input the results of previous step, and identifies the set of strategic objectives that are *indirectly affected* by each CAP. As shown in Figure 18, when a candidate Agile practice (e.g. CAP_x) *directly* contributes to a strategic objective (e.g. SG₁), and there is a relationship between SG₁ to a higher-level strategic objective (e.g. SG₂), it is a valid conclusion so say the CAP_x is indirectly affecting the SG₂, through the SG₁. This analysis is called propagative, as it is propagating the direct affects of CAP to the higher levels of SG.

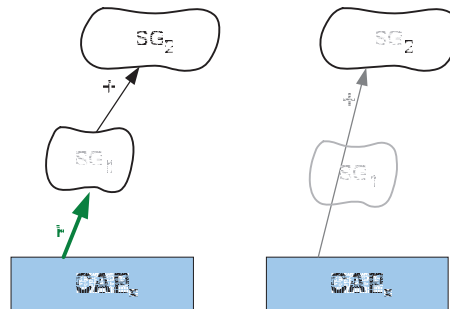


FIGURE 18: PROPAGATIVE STRATEGIC ANALYSIS: (LEFT) ORIGINAL DIRECT CONTRIBUTION; (RIGHT) PROPAGATED INDIRECT CONTRIBUTION

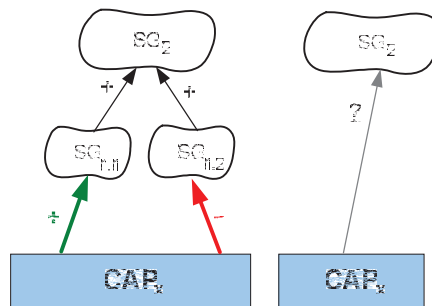


Figure 19: Propagative Strategic Analysis; a conflicting scenario






The propagative strategic analysis is not always a straight forward process. Conflicting scenarios can arise, such as the one depicted in Figure 19, in which the CAP_x is making a positive indirect contribution to SG₂ through SG_{1.1} and a negative one through SG_{1.2}. The following procedure is proposed to systematically run the propagative strategic analysis. This type of analysis is based on the *i** forward propagation procedure described in (Horkoff & Yu, 2009).

5.3.1 How to run the Propagative Strategic Analysis

Propagative strategic analysis is an iterative process, which starts from the lowest levels of the strategies graph, and in each iteration, it propagates the affects of the CAP to one higher level of the SG. The procedure keeps track of the CAP affect on every strategic objective, by assigning a *qualitative contribution tag* to analyzed objectives. The contribution tag specifies the type of contribution that a strategic objective has received from the CAP, which can be one of the following options, presented in Table 10.

At each iteration of this procedure, what will be propagated is the value of contribution tags, which have been assigned to the strategic objectives in previous iterations. The use of graphical notation for the contribution tags makes it more intuitive for the analysis team to run the procedure.

TABLE 10: CONTRIBUTION TAGS

| Contribution Tag | Graphical Notation | Description |
|--------------------------|---|--|
| Strongly Supported (SS) |  | The SG is receiving direct or indirect Strongly Positive (++) contribution from the CAP |
| Partially Supported (PS) |  | The SG is receiving direct or indirect Positive (+) contribution from the CAP |
| Partially Declined (PD) |  | The SG is receiving direct or indirect Negative (-) contribution from the CAP |
| Strongly Declined (SD) |  | The SG is receiving direct or indirect Strongly Negative (--) contribution from the CAP |
| Conflicted (C) |  | The SG is receiving direct or indirect both positive and negative contributions from the CAP |
| Unaddressed (U) | N/A | The SG is not being either directly or indirectly contributed by the CAP |

The propagation process begins by initializing the contribution tags of the strategic objectives which are being directly affected by the CAP. Then, the process continues by propagating up or forward in the direction of the links. Propagation is done using a combination of rules and user judgment. Propagation rules indicate how to propagate contribution tags through decompositional or contributinal links of a strategies graph.

Propagation through contribution links – This kind of propagation takes the contribution tag of a strategic objective and combines it with the type of contribution link, which connects it to the higher level objective. The propagation rules are specified in Table 11.

TABLE 11: PROPAGATION RULES SHOWING RESULTING TAGS, WHEN THE CONTRIBUTION TAG OF A CHILD STRATEGIC OBJECTIVE WILL BE PROPAGATED UP THROUGH A CONTRIBUTION LINK (ADAPTED FROM (HORKOFF & YU, 2009))

| Contribution Tag of SG | | Contribution Link Type | | | |
|------------------------|------------------------|------------------------|----|----|----|
| | Name | ++ | + | -- | - |
| ✓ | Sufficiently Satisfied | ✓ | ✓ | ✗ | ✗ |
| ✓. | Partially Satisfied | ✓. | ✓. | ✗. | ✗. |
| ⚡ | Conflict | ⚡ | ⚡ | ⚡ | ⚡ |
| ✗. | Partially Declined | ✗. | ✗. | ✓. | ✓. |
| ✗ | Strongly Declined | ✗ | ✗. | ✓. | ✓. |

For example, as shown in Figure 20 (A), the process starts by initializing the contribution tag of SG_1 to PD (as it is receiving a negative direct contribution from the CAP_x). In the next iteration, in order to specify the indirect contribution type of CAP_x to SG_2 , the contribution tag of SG_1 will be propagated up, and based on the propagation rules, the contribution tag of SG_2 will be also specified to be PD.

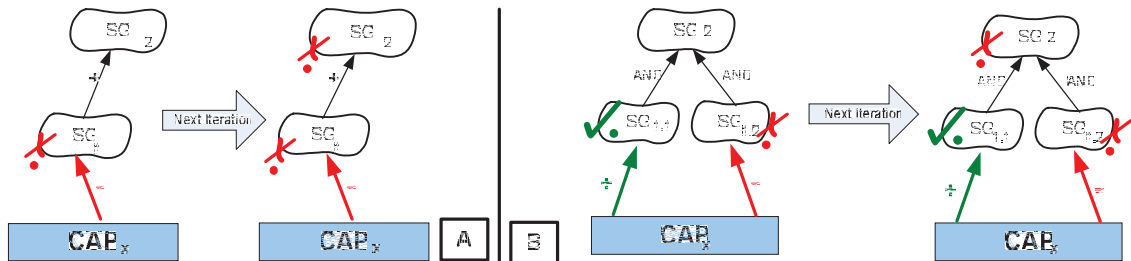


FIGURE 20: PROPAGATION EXAMPLES

Propagation through decompositional links – When propagating through AND decompositions, the values of all the children objectives are combined together, in order to specify the contribution tag of higher level objective. The decision is made using a minimum function over the children’s tags, with the following tag ordering:

$$SS > PS > C > PD > SD$$

Unaddressed labels are not propagated. When propagating through OR decompositions, the values of all children are combined using a maximum function, following the tag order above. For instance, as shown in the Figure 20 (B), when $SG_{1,1}$ and $SG_{1,2}$ are respectively initialized with PS and PD, the SG_2 will receive PD, which means that the CAP_x is making a negative contribution to it. The reason behind this decision is that when a strategic objective is AND composition of other objectives, it can be considered to be supported by a CAP only if all of its children are already supported by the CAP.

Resolving Multiple Contributions – Often multiple contribution links contribute to the same objective, leading to situations with multiple sources of incoming support tags. Multiple tags are collected in a “tag bag” for each objective. Bags of multiple tags can be automatically resolved into a single tag when all support tags are of the same polarity and at least one sufficient or strong tag is present. For example, if the bag is {PS, PS, SS} the resolution is automatically SS, using a roughly summative approach. If the evidence collected in the tag bag is partial or conflicting, user judgment is needed to combine support levels. SAAF users employ their knowledge of the business and its objectives to make a decision which implies the level of support. For example, a tag bag, {PS, PS, PS}, could be combined to produce PS or SS, depending on the cumulative level of support judged by SAAF users. Another tag bag, {PS, PD, SD}, could be resolved to C (conflict) PD or SD, depending on the relative strengths of the contributions as decided by users. Such decisions often prompt useful discussions which help to clarify the meanings of objectives or links in the SG model.

This propagative analysis can be summarized with the following steps:

Step 1: Initialize the contribution tag of strategic objectives which receive direct contribution(s) from the CAP, based on the guidelines of Table 10.

Repeat steps 2 and 3 while there is an unevaluated strategic objective

Step 2: Propagate the contribution tags one level up, based on the *automatic propagation rules*, specified the Table 11.

Step 3: Resolve the multiple contributions.

Using the propagative analysis procedure, the effectiveness of individual CAPs in supporting or declining a wider range of organizational objectives (compared to the results of contribution analysis procedure) can be assessed. For example, in Ericsson experiment, after running this analysis on the graph, which was shown in Figure 16, we inferred that the enactment of the CAP “Sprint Planning”, not only affects objectives such as “Better Visibility” and “Learning”; it will positively affect higher level strategic objectives, e.g. “Productive Individuals”.

5.4 Aggregative Strategic Analysis

The purpose of aggregated strategic analysis is to explore the overall impact of the new Agile method, which is composed of a number of Agile practices, over the strategic objectives of an organization. In this analysis, for each organizational strategic objective, all the contributions from all candidate practices of new method are combined to produce the contribution of new Agile method to that specific objective. After aggregation of contribution relations, every organizational strategic objective will take one of the following statuses:

- *Strongly Supported* – received strong positive contributions
- *Supported* – received homogeneous positive contributions
- *Conflicted* – received heterogeneous contribution types from different practices

- *Declined* – received homogeneous negative contributions
- *Strongly Declined* – received strong negative contributions
- *Unaddressed* – not contributed to by any practice, neither directly nor indirectly

The presence of unaddressed yet important objectives shows the inadequacy of the new method in addressing some of the organizational strategic objectives. Conflicted and Declined objectives highlight the controversial aspects of the new method, which might leave harmful impacts over the organizational strategies. Prospecting the status of each strategic objective helps the analysis team to modify the transition process, in such a way that maximizes the strategic benefits.

5.4.1 How to run Aggregative Strategic Analysis

Aggregated Strategic Analysis can be considered as a special case of Propagative Strategic Analysis. The only difference is that in Aggregated Strategic Analysis, the leaf nodes of the strategies graph would receive multiple contributions from different Agile practices, and thus, it is more expected to do conflict resolution at the lower-level strategic objectives. Once the impacts of multiple direct contributions to various strategic objectives are resolved, the rest of analysis can be followed as of the guidelines of Propagative Strategic Analysis.

5.5 Strategic Balance Analysis

Following Balanced Scorecards, one of the goals of the SAAF framework is to investigate whether the new Agile method makes a balanced contribution to all categories of objectives. More specifically, in this framework, the transition to a new method is considered to be unbalanced if its positive contributions to one category of strategic objectives lead to significant bad effects on some other category of objectives. The balance of a transition does not imply that the selected set of practices is the optimum set, but an optimum set should make balanced impact over the strategic objectives. Here, four techniques are described for the pre-adoption analysis. For each technique, first the target of the technique is stated, and then the steps which should be taken.

5.5.1 Balance Improvement

The introduction of a new process to an organization is often motivated by a set of strategic objectives that as their current status is not satisfactory. This can be due to the issues with the current development process of organization, which prevented the organization from achieving its strategies; or the change of strategies, which made the current process unable to meet the target values new strategic objectives. To reach a balanced state after adopting the new process, it is necessary that the new process addresses the strategic objectives that are below their desired status. Thus the first target in analyzing the balance of a process adoption initiative is to investigate whether:

The new process is positively contributing to the strategic objectives, which are expected to be improved.

The following steps are proposed to inspect the balance improvement:

1. Identify the list of strategic objectives which their indicators were planned to get improved – this set of objectives is called ToBeImprovedObjs.
2. Based on the results of *Strategic Contribution Analysis* and *Aggregative Strategic Analysis*, identify the list of strategic objectives that from the fragments of the new process will receive:
 - a. Only positive contributions – this set is called ToBeImprovingObjs.
 - b. Conflicting contributions (both negative and positive contributions) – this set is called ToBeConflictingObjs.
 - c. Only negative contributions – this set is called ToBeDeterioratingObjs.
3. To analyze the Balance Improvement target, following cases should be considered:

- a. If (ToBeImprovedObjs is the subset of ToBeImprovingObjs) then the target is met.
- b. If (ToBeImprovedObjs has any common element with the ToBeConflictingObjs) then every strategic objective which belongs to the common set should be further investigated to anticipate whether it will eventually deteriorate or improve (this investigation will be explained in the next subsection).
- c. If (ToBeImprovedObjs has any common element with the ToBeDeterioratingObjs) then the target cannot be fully met. When this situation happens, the analysis team might decide on corrective actions on the new process (e.g. altering the constituent fragments, or changing the settings of current fragment), depending to the significance of subject strategic objectives.

5.5.2 Balance Preservation

The improvement of strategic objectives that were expected to be improved by the adoption of new process, should not result in a false perception of overall process improvement. Because, the new process can be designed as to effectively improve a limited set of objectives, but deteriorate some other sets. In designing a new process the optimum solution is one which makes minimum conflict with organizational strategies, which are not at the focus of improvement. Thus, the second target in analyzing the balance in adopting a new process is to inspect whether:

The new process is not causing uncontrolled negative impacts on the strategic objectives, which are not within the focus of improvement.

The analysis technique is described in the following steps:

- 1) Calculate the *contribution weight* of every strategic objective, expressed as CW (SO) – which is the sum of the value of contribution relations that a strategic objective (SO) receives from different fragments of the new process. The value of (qualitative) contribution links is quantified as follow:
 - a. +1 for Positive Contribution
 - b. +2 for Strongly Positive Contribution
 - c. -1 for Negative Contribution
 - d. -2 for Strongly Negative

- 2) The target is ideally met when the contribution weight of all strategic objectives is positive. But that is not usually the case. Thus, for every strategic objective (SO) with the WC (SO) < 0 :
 - a. If the SO belongs to the ToBeDeterioratingObjs then the analysis team should conduct a *trade off analysis*:
 - i. Is it tolerable for the organization to see this strategic objective deteriorating, in trade off with the improvement of other objectives?
 - ii. Is there any fragment in the new process of software development, which could overcome the negative contributions, and missed at step *Strategic Contribution Analysis*?
 - iii. Is there any corrective action applicable on the new process, which would overcome the negative contributions? (If yes, the impacts of this

corrective action on other strategic objectives of organization should be re-explored by repeating step Strategic Contribution Analysis.)

- iv. If all above conditions are denied, the negative contribution of new process to the subject strategic objective is considered to be uncontrollable.
- b. If the SO belongs to the ToBeConflictingObjs then the analysis team should conduct a similar *trade off analysis*, but with the following condition added:
 - v. Is it worth accepting the negative contribution, in trade off with the positive ones towards the same strategic objective?

5.5.3 Balance Preservation across Categories

The improvement of organizational strategic objectives can be viewed at different levels. The previous technique was focusing on the instances of strategic objectives, and was concerned with keeping the balance of each one. However, one of the issues which might disturb the balance of a process adoption initiative is that the new process fully deteriorates one or more categories of strategic objectives. For instance, for an organization that is behind the desired status of its customer related strategic objectives, the first priority of process improvement is the improvement of its customer related strategies. However, this should not result in the deterioration of other categories of strategic objectives such as business or internal process. Thus, the third target in analyzing the balance in adopting a new process is to inspect whether:

The new process is not causing overall deterioration of a particular category of strategic objectives, for the sake of improving some other categories.

The analysis technique is carried out with the following steps:

1. For every category of objectives, calculate the total weight of the negative contributions (TWNC), and the total weight of the positive contributions (TWPC) made towards the strategic objectives of that category; also calculate the total weight of contributions (TWC) of that category, as $TWC = TWPC - TWNC$.
2. A process adoption is considered unbalanced if either of these conditions happen even in one category of strategic objectives:
 - a. The TWC be negative, which shows that the new process is causing the deterioration of that category.
 - b. The TWC be positive, but the TWNC be considerably low, which does not justify the overall positive contribution of the new process to this category. (the TWNC is a negative number thus the lower it is, the more negative contributions are received by the category)

5.5.4 Homogenous Contributions across Categories

The other perspective from which the balance of a process adoption can be view is to figure out the homogeneity of contributions that the new process makes to different categories of strategic objectives. This kind of balance analysis is to see whether the new process is homogenously treating all categories, i.e. different categories are benefiting and hurting to the same degree from the adoption of new process. This can be investigated by identifying the variance of the percentage of different types of contributions, made from the new process to different categories of strategic objectives. Thus, the forth target in analyzing the balance in adopting a new process is to inspect whether:

The new process is causing homogenous impacts over all categories of strategic objectives.

Analyzing the homogeneity of contributions across different categories has a precondition. This analysis is meaningful if strategic objectives be evenly distributed across different

categories; i.e. the variance of the number of strategic objectives in different categories does not exceed certain limit. For example, in an organization with 100 strategic objectives grouped in 3 categories, with distribution of 28 objectives in the first category, 38 in the second, and 34 in the third; it makes sense to analyze the homogeneity of contributions across categories, because the variance is not too much and objectives are almost evenly distributed among categories. But, if objectives were distributed e.g. as 80, 15, 5 among three categories, then due to the high variance of number of objectives across categories, the homogeneity analysis will not be meaningful. The reason is that for instance, the last category will be fully supported by the new process if it receives only five positive contributions; whereas the first category would need 80 positive contributions to get the similar status.

The following technique is proposed for this analysis:

- 1) Build the *contribution distribution table*, which for every category of strategic objective shows the number and percentage (within the category) of strategic objectives which are:
 - a. *Supported* – SO belongs to the ToBeImprovingObjs
 - b. *Denied* – SO belongs to the ToBeDeterioratingObjs
 - c. *Conflicted* - SO belongs to the ToBeConflictingObjs
 - d. *Unaddressed* – Otherwise.
- 2) To investigating the homogeneity of the contributions across categories, explore the variance of contributions of the new process to the strategic objectives. This technique would show whether the new process is making similar contributions to different categories. To do so, in distinction of the status of strategic objectives (supported, denied, conflicted, unaddressed), calculate the variances of percentages and numbers in different categories. For instance, calculate the variance of the percentage of strategic objectives with the status value of Supported among different categories. The lower the variance

measures, the more homogenous contributions are made from the new process to different categories of objectives.

Chapter 6. Strategic Actor analysis: an Approach for Identifying Process Concerns

For many software development companies, the sense of need for transitioning to Agile emerges when they observe some inefficiency symptoms in their development process; symptoms such as being late to market, over-budget development, and unhappy customers. This need will be intensified when faces the huge wave of advertisements on the success of Agile methods in bringing performance and efficiency to software teams. Thus, more and more software companies are moving to Agile methods, aiming at resolving their process problems, and reaching higher productivity. On the other hand there are numerous reports that shows Agile does not work for every project/organization; and blind transition to Agile may cause a company devastating costs, such as human training, conflict resolution, process misunderstanding, etc .

One of the early steps of process (particularly Agile) adoption frameworks is to check the adoptability of Agile with organization/projects. This typically includes a set of check lists to evaluate the capability of organization for adopting Agile method, and suitability of Agile practices for the projects which are under development. For instance, (Sidky *et al.*, 2007), while proposing an Agile adoption framework, defined an Agile measurement index, for evaluating the degree that an organization can become Agile. Based on this index, the first step that they defined in their Agile adoption framework was to identify *discontinuing factors* – indicators that their existence in an organization proves the unreadiness of organization for adopting Agile. Similar steps have been introduced by Agile method adoptions of (Qumer and Henderson-Sellers 2008) and (Sureshchandra & Shrinivasavadhani, 2008).

This framework suggests that an organization, before adopting any Agile practice, should have a clear understanding of its problems, in order to then decide on the suitability of Agile (or non-Agile) solutions for its development process. The framework does not emphasize on any particular approach for identifying process problems, as there exist a variety of such solutions, ranging from classical process assessment ones to light-weight process inspection

methods. However, the framework provides a solution for the identification of process concerns, which is based on the notion of strategic actors.

This framework suggests that an organization, before adopting any Agile practice, should have a clear understanding of its problems, in order to then decide on the suitability of Agile (or non-Agile) solutions for its development process. The framework does not emphasize on any particular approach for identifying process problems, as there exist a variety of such solutions, ranging from classical process assessment ones to light-weight process inspection methods. However, the framework provides a solution for the identification of process concerns, which is based on the notion of strategic actors.

This section describes the solution of SAAF for the extraction and analysis of process concerns. The chapter introduces a modeling notation, called Itemized Strategic Dependency, which is derived from the i* Strategic Dependency diagrams (Yu & Mylopoulos, 1994). The proposed modeling notation facilitates the process of extracting as-is concerns of an organization, by visualizing the dependencies among various organization actors, and guiding them towards the identification of related problems.

6.1 Lightweight Process Modeling for Identify as-is Process Concerns

Lightweight modeling techniques are identified as an effective mechanism for knowledge acquisition during early stages of SPI initiatives (Savolainen *et al.*, 2007). Such models can support group- or individuals- interview sessions, and help people to better express their tacit knowledge of as-is software process. In fact, this stage of framework can benefit from a wide range of variety of modeling techniques. However, the modeling techniques should be simple enough for communicating with inexpert users. This framework, inspired by (Ahonen *et al.*, 2002), recommends the following steps for identifying process concerns:

3. *Form a transition team* – that is group of individuals that can represent various organization roles. This group of individuals will be interviewed in at least two rounds, to identify and consolidate process concerns.
4. *Gain an initial understanding of development process and its problems* – this can happen by interviewing people of organization, and the use of simple process and

data models that will be drawn on a whiteboard. Predefined questionnaires can be quite helpful in structuring the interview sessions and clarifying different process activities, artifacts, roles, responsibilities, and perceived problems. An example of such questionnaires can be found in (O'Donnell & Richardson, 2008).

5. *Develop electronic version of process model* – based on the results of the previous step, the perspectives of different people will be integrated to develop the generic model(s) of current process. Indeed, the expressed process concerns will be analyzed to come up with a list of common concerns.
6. *Consolidate the initial understanding* – the purpose of this step is to ensure that a right perception of development process and its problems have been achieved. In another round of interviews, the electronic models of process will be shown to the people of organization, asking for their opinion about the correctness and completeness of the models. These models would also help the interviewees to raise some other process concerns, which were missed at the initial round of interviews. Besides, the list of previously identified process concerns will be discussed with interviewee(s) (those concerns that are relevant to their role), ask for their idea about the validity, severity, and priority of each concern.
7. *Finalize the list of process concerns* – this is just to come up with a list of all process concerns mentioned by different interviewees. This list will be analyzed in the next step of framework to explore the cause-effect relations of process concerns, and evaluate their impact over the strategic goals of organization

6.2 Strategic Dependency Models

6.2.1 Strategic Actor

The *i** modeling framework defined the notion of strategic actor as any kind of active entity that performs activities to achieve its goals by exercising its know-how (Yu & Mylopoulos, 1994). Regarding the purpose of this work, the framework interprets the strategic actors as individuals involved in the process of software development each carrying out certain Agile practices to achieve particular strategic objectives. For instance, in order to achieve the *strategic objective* “Clear Understanding of Requirements”, a *strategic actor* “Project

Manager” might decide to incorporate the *Agile practice* “Develop Formal Requirement Specifications”. A strategic actor might fulfill its strategic objectives by itself, or by depending on other actors.

6.2.2 Strategic Dependency Diagram

In order to visualize the dependency relations of strategic actors, the *i** framework introduced the Strategic Dependency (SD) diagrams. The SD diagrams can show how one actor may depend on other actors for tasks to be performed (Task Dependency), goals to be achieved (Goal Dependency), and resources to be provided (Resource Dependency). For instance, Figure 21 shows a sample *i** SD model with two actors: “System Manager” and “Designer”. It visualizes that the Systems Manager depends on Designer to design the system and write down the implementation proposal, where the earlier is a goal dependency as designers can chose by themselves how to design the system. Whereas writing down the implementation proposal is a specific and clear-defined task. This sample model shows that Designer depends the other way on the Systems Manager for Requirements Document. (This model has been adopted from our case study in R&D UNIT.)

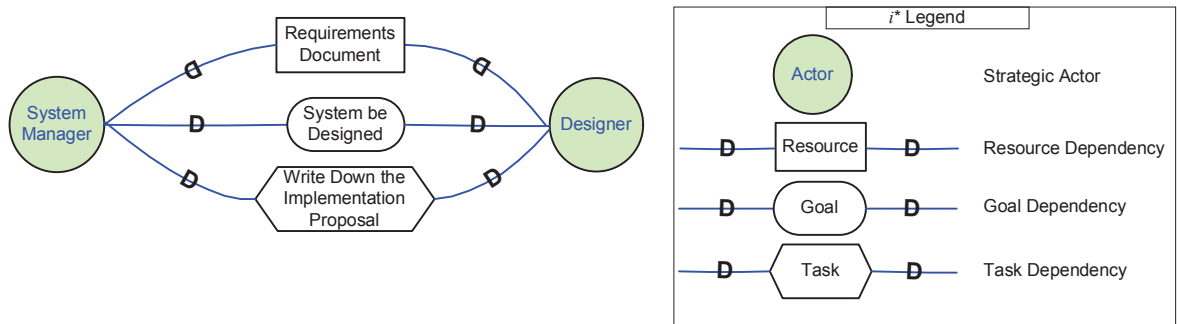


FIGURE 21: SAMPLE I* STRATEGIC DEPENDENCY MODEL

Regarding the purpose of this step of framework (modeling dependency relations to extract process concerns) the original *i** SD diagrams could have been used. But, due to the following reasons, the framework needs a simpler dependency models:

- Not everyone could understand/use SD diagrams – It is not easy for people who are not familiar with the i^* modeling framework to communicate through SD diagrams. The main problem is that untrained people cannot easily distinguish among various dependency types.
- SD diagrams can become way complex – The original SD diagrams depicts one typed dependency link for every dependency relation. The positive aspect of this approach is that the resultant model has a rich semantic meaning; however, as the number of dependency relations increases, the complexity of models intensifies.
- Original SD diagrams are over-expressive – regarding the purpose of this step of framework, original SD diagrams provide semantic information that is not needed. For example, there is no need to make a modeling distinction between resource and task dependencies, when we just want to know the set of dependency relations between two actors.

6.2.3 Itemized Strategic Dependency

(Chiniforooshan *et al.*, 2010) introduced a variant of the i^* SD diagrams, called *Itemized Strategic Dependency* (ISD), which has a simpler notation for visualizing the dependency relations. In ISD diagrams a single link will be used to represent all of the dependency relations of one actor to another one, and objects of dependency will be all listed on that link. Figure 22 shows a sample ISD diagram. Such models, by decreasing the number of dependency links, reduced the complexity of resultant models, and made them easier for communicating with people who are not familiar with the i^* . For example, if the dependency model of Figure 22 was developed in original SD diagrams 14 dependency links had to be used – each having an underlying semantic concept.

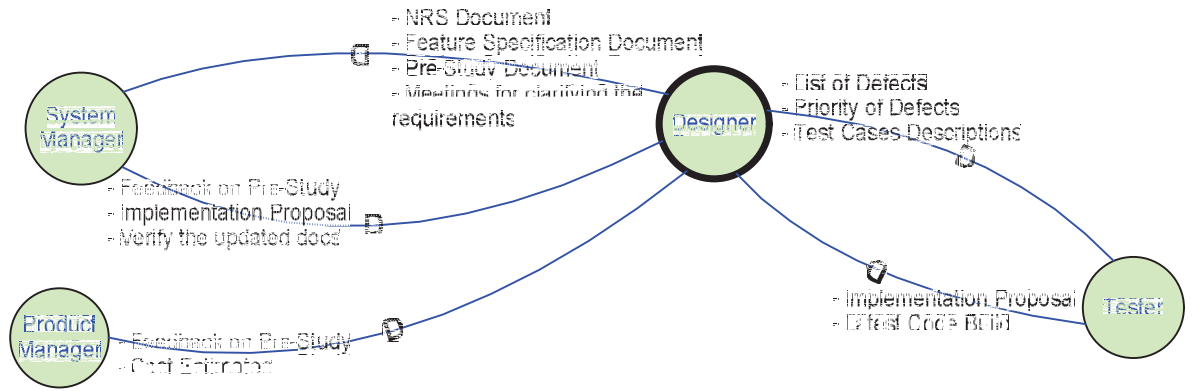


FIGURE 22: PART OF FUNCTIONAL ISD, DEVELOPED IN ERICSSON PROJECT

One of the improvements of ISDs over SD diagrams is the introduction of *Viewpoint* actor, which is the actor from whose perspective the dependency model is developed. This actor is depicted in bold circle notation. Developing various ISDs from the viewpoint of different actors helps to reveal the misunderstandings of different organizational actors from their responsibilities and expectations. ISD models are in two types: *Functional ISD*, and *Quality ISD*.

6.2.3.1 Functional ISD

As defined in (Chiniforooshan *et al.*, 2010) a functional dependency is one which does not deal with the qualitative aspects of the dependency objects. An ISD that represents only the functional dependencies of its viewpoint actor is called Functional ISD (FISD). Figure 22 shows one of the FISDs that has been developed in Ericsson project. It visualizes the basic things that a Designer depends on others to be provided, and vice versa. However, this model does not deal with the quality attributes of the expected things. For instance, it shows that a designer expects the System Manager to provide the NRS Document (Node Requirements Specification).

6.2.3.2 Quality ISD

An ISD that represents the quality attributes of functional dependencies is called Quality ISD (QISD). Associated quality attributes of each functional dependency (if there would be any) are listed below the object of (functional) dependency. Figure 23 shows a QISD which complements the FISD of Figure 22. As shown here, QISDs provide more detailed information

about the objects of dependency. For instance, Figure 23 shows that while Designer expects System Manager to provide the NRS Document, the document is expected to have certain quality attributes, e.g. be adequately detailed, technical, also clear and understandable.

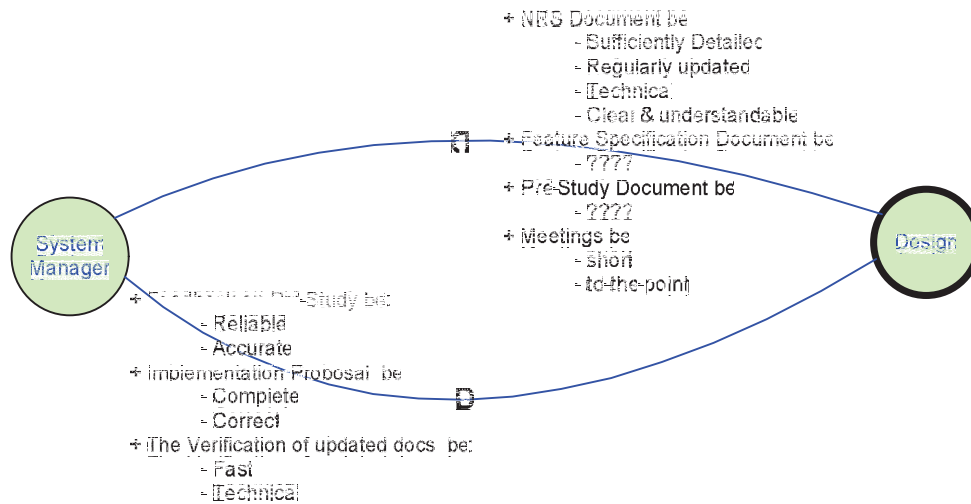


FIGURE 23: PART OF QUALITY ISD, DEVELOPED FOR R&D UNIT PROJECT

6.3 Identifying process concerns using ISDs

The intention of using dependency models in the process of concerns identification is to bring these two issues into the attentions of interviewees: “*what are they expecting from others?*” and “*what are they expected to provide for others?*”. People might have different perspectives regarding the responsibilities of their own and other organization members. These perspectives, when expressed as dependency relations, can be formulated at two levels:

1. Functional Dependency: that is intended to express on what matters (goals, tasks, resources, etc.) organizational actors depend on each other. These dependency relations can be formulated as “*A depends on B for Φ* ”, where A and B are two organizational actors, and Φ is a set of goals, activities, or resources that A depends on them and expects from B.

2. **Quality Dependency:** provides deeper details of the dependency relations, which are related to the quality of dependency objects. These dependencies can be formulated as “*A depends on B for Φ with Ψ* ”, where Ψ is a set of quality attributes expected from different objects of dependency that belong to Φ .

Process and data model that are associated with actor models can be used to clarify the functional responsibilities and expectations of each actor. But, these models do not deal with the quality attributes of the work items, especially from the viewpoint of those whose activities depend on the provision and quality of those work items. Therefore, this framework argues that the use of ISDs can help to better understand the current process and identify its concerns. The following guideline states how to incorporate the ISDs into the generic process of concerns identification.

1. *Gain an initial understanding of development process and its problems* – while asking generic questions about the development process and its concerns, guide interviewees to express the things for which they depend on other organization roles, and the things they perceive others expect from them.
2. *Develop electronic version of process model* – based on the results of the previous step develop the Functional ISD from the viewpoint of different actors.
3. *Consolidate the initial understanding* – during the second round of interviews, each interviewee will be shown the initial FISDs that were developed from his/her viewpoint. Then participants are asked to express the following information:
 - a. *The validity of the identified dependency models* – interviewee is asked to confirm if the visualized FISDs correctly and completely represent the functional dependencies among him/her and other organization actors.
 - b. *The quality attributes expected from each functional dependency* – interviewee is asked to express his/her expected quality attributes from the functional dependencies to other actors. The collected information will be used to develop Quality ISDs.
 - c. *The status of functional and quality dependencies* – interviewee is asked to evaluate the extent at which functional and quality dependencies are satisfied. Functional dependencies are often evaluated in Boolean, while

quality dependencies can take a range of values, such as (Fully Satisfied, Partly Satisfied, Partly Denied, Denied).

4. *Finalize the list of process concerns* – by integrating different FISDs and QISDs, the overall dependency models of organization and a unified list of process concerns will be developed.

6.4 What kinds of process concerns are identifiable with this approach

The use of ISD models can help to identify the following types of process concerns:

1. *Missing Quality Attributes*: The first and foremost process concerns that are identifiable with ISD models are missing quality attributes. These concerns are often expressed by interviewees when they are exposed with their initial FISDs, and asked to express their expected quality attributes. From what we observed in our experience with R&D UNIT, most people first express the quality attributes that they had expected from specific dependency objects, but were never or barely achieved. For instance, when we showed the FISD of Figure 22 to a designer, and asked about the quality attributes of the NRS Document, he instantly expressed his unhappiness about the ambiguity, incompleteness, and superficiality of that document.
2. *Missing Dependencies*: The second type of process concerns that can be identified with ISDs are those which are due to missing dependencies, i.e. dependencies that should have been exit for facilitating the work. These dependencies might be in form of functional dependencies with the already existing organization actors, or some other actors that their existence could be useful.
3. *Extra Dependencies*: the other type process concerns that can be identified by ISDs are those that are due to unnecessary collaborations and dependencies. These are often expressed as overheads activities to the process of software development, which do not add any tangible value, or in the best case their value is not recognized by the people of organization.

6.5 Analyzing the Identified Process Concerns

As (Florac *et al.*, 1997) states “Identifying process issues is like the tip of the iceberg”. After the identification of process concerns the more important step is to analyze them, in order to identify their causal relations. In general, the cause-effect analysis is the most commonly used approach for the analysis of process concerns. It starts by the agreement on the existence of a problem, and then through a series of “Why” questions, tries to identify the root-cause of the problem. On the other hand, the approach tries to identify the potential effects of the identified problem. Fishbone diagram is one of the most common approach for cause-effect analysis in quality management, which identifies causes and effects of a problem and sort them in relevant categories (Tague, 2004).

A number of guidelines have been so far proposed for the analysis of process concerns. (Florac *et al.*, 1997) proposed a measurement-based approach in which after the clarification of process concerns, a set of appropriate measures will be defined in order to analyze the variability of concerns. It also focuses on the identification and the control of processes that encompass the concern. (Gorschek & Wohlin, 2004) proposed a method for the analysis of process concerns based on their dependency relations and priorities. The CMMI proposes a process area (at maturity level 5), named Causal Analysis and Resolution (CAR), which provides a set of generic guidelines for determining causes of process concerns, analyzing those causes, and taking appropriate actions to address the causes (SEI, 2006). The CMMI also proposes another process area (at maturity level 2), named Measurement and Analysis (MA), which deals with the definition of indicating measures for the handling process concerns (as well as other kinds of information for which measurement is need).

6.5.1 Strategic Process Concern Analysis

Strategic Process Concern is one of the SAAF procedures, proposed to investigate the impacts of current process concerns on the strategic objectives of an organization This framework suggests the following steps for the analysis of Identified Process Concerns (IPCs): *Categorization, Strategic Cause-Effect Analysis, and Agile Practices Capability Analysis.*

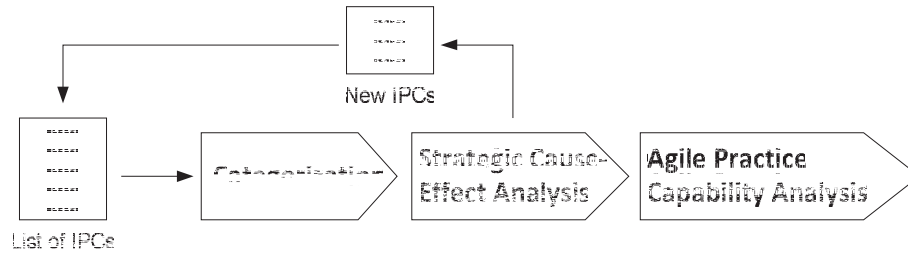


FIGURE 24: MAJOR ACTIVITIES AND ARTIFACTS OF CONCERN ANALYSIS

6.5.2 Categorization of IPCs:

The purpose of this step is to group related process concerns, in order to facilitate the further monitoring and reporting activities of the framework. Process concerns can be categorized in various ways, and the framework is quite flexible on the choice of technique for categorization of concerns, as well as the ultimate concern categories. However, the framework suggests a two-level categorization of IPCs, where concerns are first categorized based on the major software engineering activities (e.g. Requirements Engineering, Implementation and Test, Project Planning and Management, etc.), and then the IPCs of each category will be further grouped into related *Areas of Concern*. For instance, for the Ericsson case, for the category of “Project Planning and Management”, the following areas of concern have been identified: "Planning based on early estimations", "Planning for multiple development iterations", "Test Planning ahead", "Task Prioritization", and "Employee satisfaction". Figure 25 shows the IPCs categorized under the area of “Planning based on early estimations”.

The most similar concept to *IPC Categorization* can be found in SPI frameworks, under the term of *packaging SPI issues*. However, it should be mentioned that SPI issues are not just the process problems (Concerns), and they can cover the potential process improvement activities. Indeed, it should be mentioned that the purpose of packaging SPI issues is to organize the implementation SPI issues, whereas the purpose of IPC categorization is just to facilitate the further analysis activities. (Gorschek & Wohlin, 2004) proposed an issue-packaging technique, which acts based on the *priority*, *dependency*, and *cost* of issues. It first identifies the *priority* and the *cost* of each issue (the cost of an issue can be demanded time and resources for its implementation); and then it uses *weighted dependency diagrams* to specify the correlations of issues. The dependency models help to categorize issues into ordered packages,

where higher priority packages containing the fundamental issues that their implementation would facilitate others. (Pettersson *et al.*, 2008) while defining a light-weight SPI framework, has introduced a similar approach (use of prioritized-weighted dependency models) for packaging SPI issues.

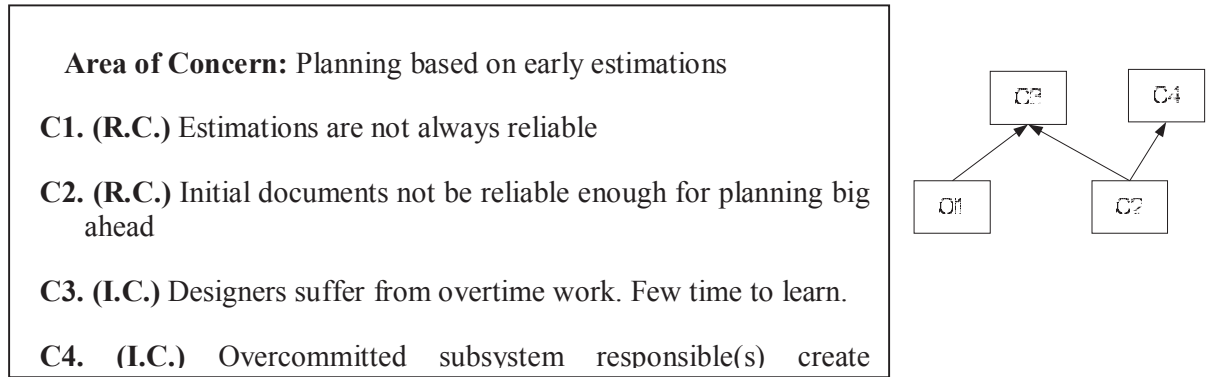


Figure 25: (Left) Sample Area of Concern, its Root Causes (RC), Intermediate Causes (IC), and Affected Strategic Goals; (Right) the cause-effect model of mentioned process concerns

6.5.3 Strategic Cause-Effect Analysis of IPCs:

The purpose of this step is to identify the potential effect(s) of each IPC over the strategic goals of organization. The basic assumption here is that “*any IPC is a cause that negatively affects some of the strategic goals of organization*”. This framework suggests a model-based approach for identifying these cause-effect relations. It first identifies the cause-effect relations among the IPCs of each category; and then visualizes the potential negative impacts of each IPC over Strategies Graph.

6.5.3.1 IPC cause-effect modeling:

Among the identified process concerns of each category, some concerns might actually be the causes of some others. Therefore, IPCs (as all were initially assumed as causes) are classified into *Root Causes* (RC) and *Intermediate Causes* (IC), where each IC can be the effect of one or more RCs. Root causes can directly hurt the strategic goals of organization, also indirectly through the ICs that they had caused. It is possible that an IC be the root of (or

impacts the emergence of) some other ICs, but it will not be classified as an RC. The reason is that the later analysis stages of framework need process concerns to be classified into disjoint sets. Figure 25 (right) shows the cause-effect diagram of the process concerns categorized under the area of “Planning based on early estimations”. Detailed cause-effect analysis can be conducted over all of the already identified process concerns, in order to further extract root-cause concerns. As shown in Figure 24 the newly identified process concerns will be fed to the analysis stage for categorization and strategic analysis.

6.5.3.2 IPC Strategic Modeling

As mentioned earlier, each IPC would affect a number of strategic goals. The purpose of this step is to pinpoint the affected strategic goal(s) of each IPC. This framework proposes the following model-based technique for this purpose:

1. Select appropriate participants for IPC analysis workshop
2. Make the already-made Strategies Graph visible to everyone, and make sure that it is correctly understood by all participants. (for instance, the SG can be projected over a big screen, or distributed in paper sheets)
3. For every IPC:
 - a. Visualize it as a text-box next to the Strategies Graph
 - b. Ask participants to identify the strategic goals that might be affected, and the intensity of negative impact (e.g. the IPC can cause the complete denial of SG, strongly hurt it, or just hurt it).
 - c. Draw the weighted contribution link from the IPC box to the affected goal.

Figure 26 shows the visualization of part of IPCs over a portion of Strategies Graph in Ericsson case.

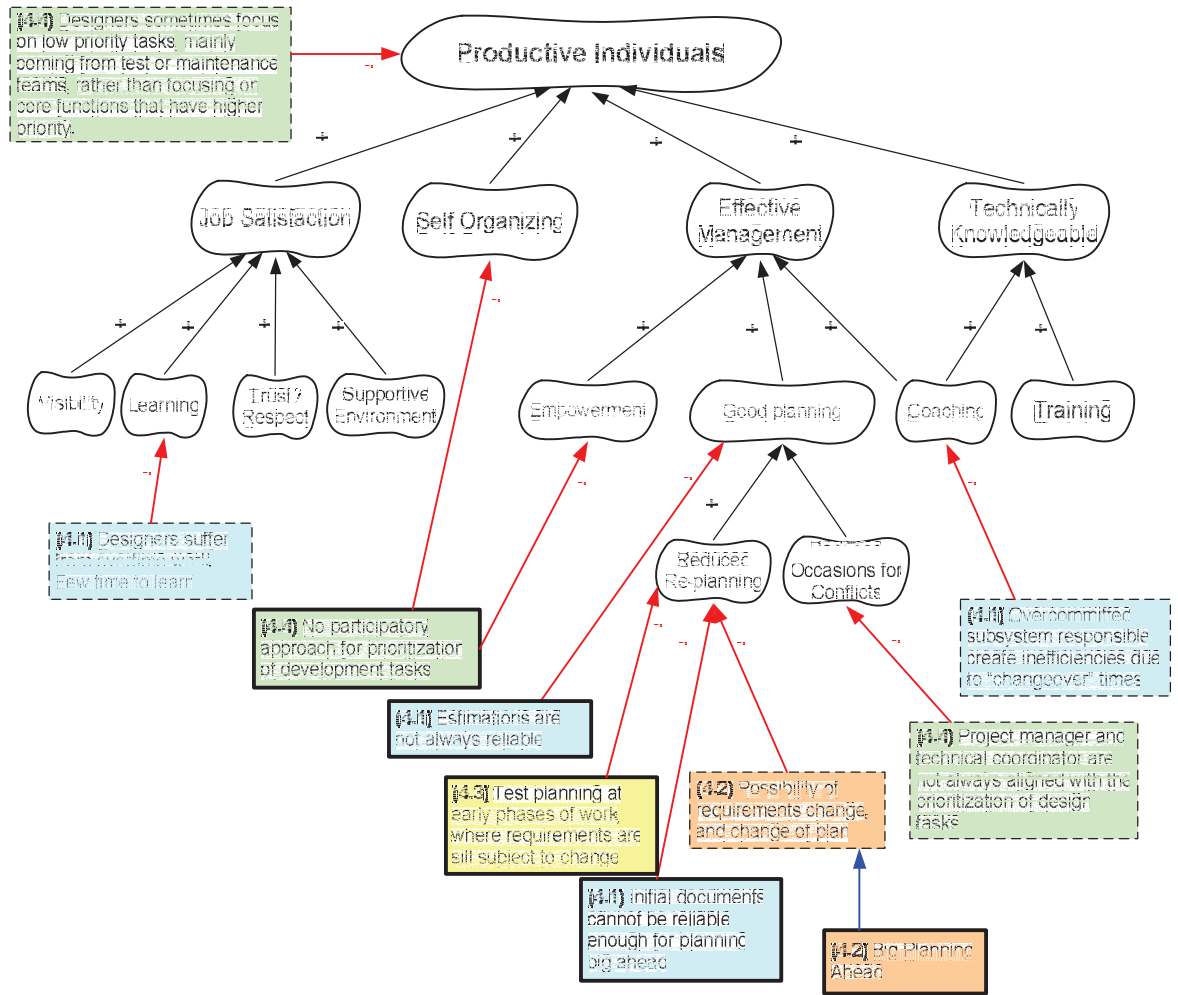


FIGURE 26: VISUALIZATION OF PROCESS CONCERNS OVER STRATEGIES GRAPH

The modeling framework proposes two types of contribution links for representing the negative impact of IPCs over strategic goals: Strong Negative (--) and Moderate Negative (-). A moderate negative link from an IPC towards a strategic goal implies that the emergence of that IPC would hurt the achievement of the strategic goal, however the strategic goal will not be severely damaged. But, a strong negative link from an IPC towards a strategic goal implies that the emergence of that IPC would almost block the achievement of the strategic goal. Further detailed evaluations could have been applied for specifying the relative impact of an IPC over strategic goals, e.g. numerical values, but the following reasons the framework chose the simple two-value links: first, to facilitate the use of framework in IPC analysis workshops;

and second, to avoid shifting the focus of participants from the identification of affected strategic goals towards the evaluation of each negative impacts.

6.5.3.3 Agile Practices Capability Analysis

This step is intended to explore the capability of every candidate Agile practice in addressing the identified process concerns of the organization. The following steps are proposed for running this analysis:

- Repeat the following steps for any of the candidate Agile Practices
- Make sure all members of transition team have similar understanding of the Agile Practice (AP)
- For Any of the IPCs
 - Explore if the AP can positively address the IPC. If it can, record the argument why the AP would positively address the IPC.

This step of framework requires a good understanding of the candidate agile practices, in terms of their capabilities and limitations. Therefore, the presence of an Agile consultant can facilitate this step. The result of this step will help managers gaining an understanding of the capabilities of proposed Agile practices in dealing with the current process problems.

6.5.4 Expected Results of Strategic Concern Analysis

The strategic concern analysis has in the following outcomes, and guidelines for the process transition team:

TABLE 12: OUTCOMES OF STRATEGIC CONCERN ANALYSIS AND ITS GUIDELINES FOR
PROCESS TRANSITION TEAM

| Outcome of strategic concern analysis | Guidelines for the Process Transition Team |
|--|--|
| Which strategic objectives have been affected by current IPCs? | Are these objectives being any supported by the new candidate Agile method? |
| Which strategic objectives received most number of negative impacts from current IPCs? | Are these objectives being strongly supported by the new candidate Agile method? |
| Which IPCs have made most number of negative impacts to the organizational strategic objectives? | Has the candidate Agile method any solution for addressing these concerns? |

Chapter 7. SAAF in Practice – An Industrial Case Experience in Ericsson

From September 2009 to August 2010, the author had been involved in an Agile Adoption initiative in one of the Ericsson R&D units in Italy. This initiative was motivated by the emergence of some inefficiency symptoms warning unit managers about their software development process. As an initial solution, the unit was encouraged (by organization process consultants) to adopt the Scrum Agile method, which had experienced success in some other R&D units of the company.

The SAAF helped managers in the organization investigate the potentially positive and negative impacts of candidate Agile practices on their strategic goals. The result of the analysis helped managers to identify customizations needed to adapt candidate practices for the organization, as well as organizational changes needed to prepare for adoption of those practices. The analysis results also helped middle managers overcome their doubts and hesitations in adopting new practices. Although the framework was developed in the context of Agile adoption and transition, it provides a framework for strategic decision making more generally, which can be used for other initiatives.

7.1 Demographic Information

The R&D unit was working on a solution product for use by telecom regulators, which functions as an intermediate knowledge management system in mobile networks. The unit consisted of about 20 members, including developers and unit managers. There were three teams of developers, each with one lead designer and three coders; The testing team consisted of one test manager and four testers. Responsibilities for architecting the system and defining the requirements were covered by three roles: the System Manager, the Solution Manager, and the Product Manager. In addition, a Line Manager was responsible for human resource

management; and the project R&D Manager was in charge of the project and was the primary connection to higher level management.

The operative level of R&D people can be classified into the following groups:

- 8 people in design/implementation
 - 3 in Database Subsystem
 - 3 in Mediation Subsystem
 - 2 in Front-End Subsystem
- 6 people in Testing
 - 1 Test Leader
 - 5 Testers

7.2 Timeline

The following timeline shows the major events of Agile adoption in the target R&D unit:

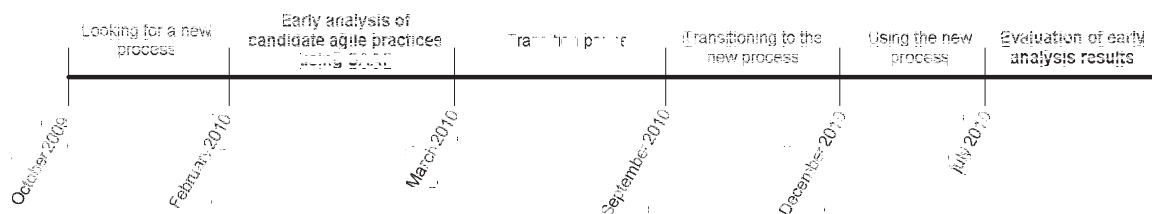


FIGURE 27: AGILE ADOPTION TIMELINE IN TARGET ORGANIZATION

As shown in the timeline, the process of transitioning to an Agile method began in October 2010, when the external process consultants of the R&D unit proposed a set of Agile practices to the organization management. In February 2010, the SAAF was used to analyze the

potential strategic impacts of the proposed Agile practices. The transition process was then paused for about six months. In September 2010, the target organization restated the process initiative and gradually moved to the new Agile method. From December 2010, the organization has been developing software using Agile process.

For evaluating the SAAF results we required the organization to stabilize its new development process, and acquire enough experience with its benefits and drawbacks. Therefore, we conducted the evaluation process about one year after the introduction of new process to the organization.

7.3 Candidate Agile Practices

The R&D unit had the intention of deploying the following Agile practices into its development environment:

- Scrum Team
- Sprint Planning
- Daily Scrum Meeting
- Potentially Shippable Increment / Short Release

These practices have been mainly adopted from the Scrum Process Model (Schwaber, 2004). The following subsections briefly describe these practices, which will be also called Candidate Agile Practices (AP).

7.3.1 Scrum Team

Scrum Team was the first candidate AP which was the subject of this study. Unlike other Agile methodologies that mainly propose technical practices, Scrum is mainly composed of a set of managerial APs, which are about the arranging people, handling requirements, and planning releases. Therefore the integral part of Scrum methodology is the team structure that it proposes for software organization. In a Scrum environment there are three primary roles:

ScrumMaster, Product Owner, and the Team. Here, we focus on the Scrum Team, which is typically characterized as follows:

- A group of individuals, *working together* to achieve sprint goals
- Scrum team is usually 6~10 people
- Scrum team does not include any of the conventional software engineering roles, such as programmer, designer, tester, architect, . . .
 - This does not mean that people must perform tasks for which they have no expertise. In fact, in most cases individuals will spend most of their time working on disciplines they worked before adopting Scrum.
 - But, individuals are expected to work beyond their preferred disciplines, whenever doing so would be of interest of the team.
- Scrum team members are expected to be
 - Self-organizing
 - Cross-functional

7.3.2 Sprint Planning (SP)

The other AP which was analyzed in this study was the Sprint Planning meeting. Almost all Agile processes have an iterative process model, in which the actual development happens in a number of cycles. In Scrum, each development cycle is called a Sprint, which is recommended to be 2 to 4 weeks. Sprint planning meetings happen at the beginning of every sprint, in order for the team to decide on the details of the Sprint's work items. This meeting is attended by players of all Scrum roles, i.e., Product Owner, Scrum Master, Scrum Team, also interested management and customer representatives. The following instructions are suggested for this meeting:

- The meeting is performed at the beginning of each Sprint.

- Product Owner explains the high priority User Stories (USs) that he would like to be implemented in the sprint.
- Team discusses with Product Owner to fully clarify and understand USs.
- Scrum team and the product owner define a sprint goal. At the end of sprint, success will be evaluated based on the degree of achievement of sprint goals.
- Team fills out the Sprint backlog, by breaking down the selected USs into tasks, and estimating the completion time of each task.
- The meeting is four to eight hours.

As-is planning approach in R&D unit

The pre-existing planning approach in the R&D unit follows the classical activity-based project management style. First the project tasks (including analysis, documentation, programming, test, etc.) are identified. Then the project manager, based on the estimated cost and priority of all tasks, schedules the tasks. In this process, the R&D unit line managers are responsible for allocating tasks to the people of the unit.

Benefits of current planning approach:

- All tasks are planned ahead, thus a big picture of overall project status and estimates of completion of each task (CRs, NRS items, Quick Studies . . .) is always available.
- The responsibilities of individuals are planned ahead, thus, theoretically, everyone knows what the project manager would expect from him/her.
- Availability or unavailability of resources is clear, for long run. Thus, the project manager can give reliable estimates to business people about the completion time of projects.

Risks of as-is planning approach:

- Estimates are usually unreliable, so the planning is always performed with an uncertainty factor
- Big plan ahead is always subject to re-planning. This issue is more critical when we provide long-time plans for development teams, because there is always something new about initial requirements, design, estimates, etc., that changes the flow of development.
- Unbalanced assignment of responsibilities (load balancing) among all team members.

Comparison of the current planning approach with SP of Scrum:

- Current planning is activity-based, while Scrum is based on fixed-time iterative planning.
- Scrum advises a participatory approach to planning, in which all of the team members participate. Current planning is performed as a managerial activity.
- Scrum planning is based on an open product backlog, which can be easily updated by team members. This kind of open backlog helps team members to have a clear understanding of project progress and the remaining tasks. In the as-is situation requirements backlog in R&D unit is not open to everyone, and the project progress is not visible to them.

7.3.3 Daily Scrum Meeting (DSM)

Increased frequency of informal communication is one of the common goals of all Agile methodologies. To this end, a variety of practices have been proposed. One of the most successful ones is the Daily Scrum Meetings. Such a meeting, as the name suggests, occurs every day between the Scrum master and Team members. Anyone else who might be interested can also attend. The following instructions have been proposed for such meetings:

- Time limit: 15 minutes
- Daily standup meeting, ideally in the morning

- Primary intention is to disseminate status information
- Inspecting the progress in relation to the planning, and making necessary adjustments (Burn-down charts can be helpful)
- Each member provides answers to the following three questions:
 1. What did you do yesterday?
 2. What will you do today?
 3. Are there any impediments in your way?
- Any impediments that are raised become the Scrum Master's responsibility to resolve as quickly as possible. Impediments such as:
 - I still haven't got the software I ordered a month ago.
 - I need help debugging a problem with _____.
 - I'm struggling to learn _____ and would like to pair with someone on it.
 - I can't get the vendor's tech support group to call me back.
- DSM should not be used for
 - Problem solving or issue resolution. Issues that are raised are taken off and then addressed by the relevant sub-groups
 - Status update in which a boss is collecting information about who is behind schedule. Rather, DSM is a meeting in which team members make commitments to each other. If a programmer stands up and says "Today I will finish the data storage module" everyone knows that in tomorrow's meeting he will say whether or not he did finish.

7.3.4 Potential Shippable Product (PSP) & Short Release (SR)

One other common aspect of many Agile methodologies is the incremental software development, which is achievable by the frequent and short-time releases of the product. This will enable a shorter feedback cycle from the customer, and facilitates the primary promise of Agile in welcoming change request, anytime during the project. Potential Shippable Product is a concept introduced by the Scrum, which motivates development teams to make an increment of the product, which can be shipped to the production site at the end of every sprint. Such an increment of software should be thoroughly tested, well-structured, well-written, and its user documents complete.

As-is product release-plan of the R&D unit:

The R&D unit had two release plans: Internal and External. The internal releases were about releasing product builds, developed by designers, to the test team. This kind of release was happening once every three weeks. The external releases were when the new version of product was being shipped to the customer site. The frequency of customer releases was twice a year. However, in some cases, e.g., high-priority change requests, the new version of software (after fixing the trouble reports, or applying the change requests) would be sent to the customer.

Risk associated with the as-is release planning:

- Late feedback from the customer
- Late detection of development issues, mainly those related to the Target Environment.
- Change of the point of contact in customer site (i.e., the person who ordered a set of change requests was no longer in charge, and the new person was not aware of previous orders)

Comparison of the as-is release planning and that of Scrum:

The iteration planning of Scrum suggests selecting a subset of items from product backlog whose development results in an increment of product, which is potentially shippable to the

customer site. In contrast, the current iteration planning of design team is targeted mainly towards the internal releases, not the customer releases.

7.4 Initial Agile Adoption Framework

The initial approach of the R&D unit for moving to Agile was inspired by the framework of Szalvay *et al.*, (2008) for transitioning to Agile. This framework is based on the use of a pilot project, in which the new Agile method will be applied, in order to identify its potential problems and possible solutions. Figure 28 shows the major steps of this framework, for transforming a non-Agile enterprise to an Agile one.

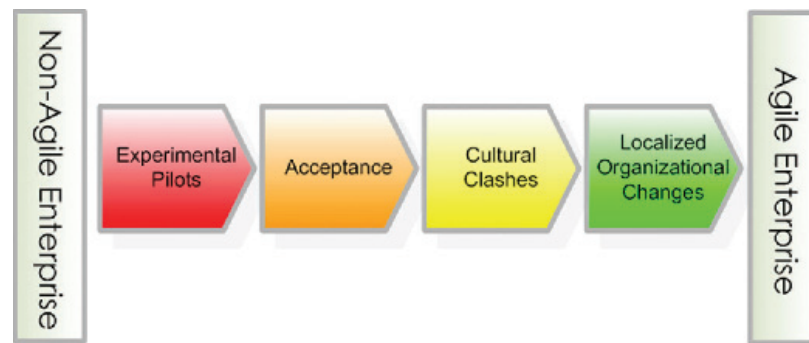


FIGURE 28: INITIAL SOLUTION OF THE R&D UNIT FOR ADOPTING AGILE (SZALVAY ET AL., 2008)

The pilot trial approach has the advantage of simplicity and speed. But, even after running the pilot trial of the new Agile method, unit managers were still uncertain about the suitability of the new Agile method for their development environment. The main reason for concern was that the pilot project was a very limited representation of the actual organization and its real projects. Furthermore, the issues (cultural, technological, human) identified at the pilot project could not fully represent the major organizational concerns.

More specifically, unit managers had the following concerns regarding the candidate Agile process:

1. Are the proposed Agile practices in line with organizational strategies?
2. To what extent will the proposed Agile practices solve the current development issues?
3. Are the inherent promises of the new Agile method attainable in the specific settings of the R&D unit project and environment?

As part of the author's research, SAAF was applied in the R&D unit to help managers find answers to the above questions in a systematic and repeatable way. The following sections describe the details of applying SAAF in this industrial setting.

7.5 Experiences with Setting up the Strategies Graph

In the Ericsson case, the R&D unit had no in-place strategic model, so as one of the earlier steps of SAAF, the author helped the organization to develop their strategic model. The first version of the Strategies Graph was developed jointly by the author of the thesis and the process manager of the organization. The author was on-site for the purpose of the study. The initial version of the SG was then presented to the R&D unit managers (in three one-hour meetings held separately with each manager) to obtain feedback and to update the graph. Then, comments from the unit managers were combined to complete the SG. Afterwards, the SG was presented to the higher organization managers of Ericsson in a public meeting with unit managers. By collecting the final remarks from all of the managers, the developed SG was considered as confirmed and the analysis process proceeded to the next step. Figure 29 shows a portion of the resultant Strategies Graph.

The development of the Strategies Graph, as mentioned in (Kaplan & Norton, 1996a), happened in both top-down and bottom-up directions. Higher organization managers were more concerned with higher strategies, whereas unit managers were more focused on actionable strategic objectives that appear lower in the graph. The Strategies Graph was

recognized by the managers as an effective mechanism for communicating organizational strategies and analyzing organizational initiatives.

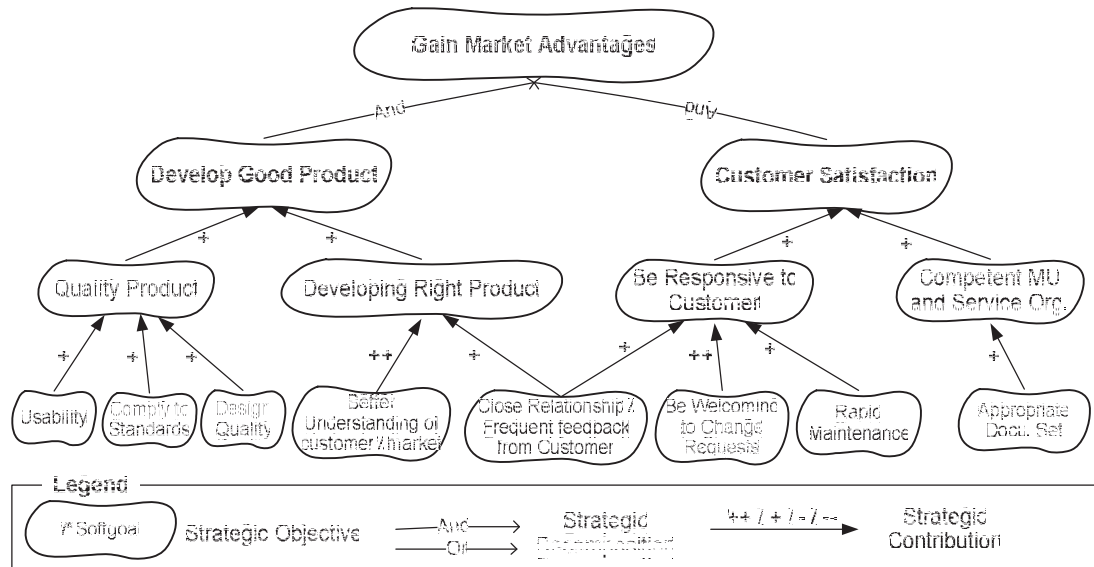


FIGURE 29: A PORTION OF THE STRATEGIES GRAPH (SG) DEVELOPED IN THE ERICSSON CASE

7.6 Experiences with Candidate Agile Practices & Retrieval of Strategic Knowledge

The Ericsson R&D unit was motivated to introduce the following Agile practices into its development process: “Scrum Team”, “Sprint Planning”, “Daily Scrum Meeting”, and “Short Release”. In addition, they were advised by an external consultant to change their team structure based on the definition of “Scrum Team”. The strategic knowledge of candidate Agile practices was used at two stages of SAAF:

1. **Completing the Organizational SG** – As mentioned before, the initial version of the SG was developed based on the major strategic concerns of the organization; However the major strategic concerns were generally abstract, and in need of

clarification as lower-level and more tangible objectives. In cases where the major strategic concerns of the organization match up with the major objectives of Agile practices, the catalogue of minor objectives were selectively used to complete the SG of organization. In this experience, the following set of major Agile objectives (from the knowledge base) were among the major strategic objectives of the organization: [*Improved Communication, Collaboration, Effectiveness, Job Satisfaction, Team Management, Improved Project Management, Improved Project Planning, and Customer Satisfaction*]

2. **Identifying the Evidential Contributions of CAPs to the SG** – After updating the organizational SG, strategic objectives matching the CAPs’ objectives were identified. The possible evidential contributions of each CAP to the objectives were retrieved from the repository and added to the SG. For instance, for the minor objectives of “Improved Communication”, six of them were contributed to by “Daily Meeting”; three by “Sprint Planning”; and one by “Short Release”.

7.7 Experiences with Strategic Analysis Process

In this research, after constructing the strategies graph, we conducted an assessment of the as-is process, in order to first identify the current process concerns of the organization, and then figure out their root-causes. Using the SG, we extended the root-cause analysis of identified process concerns to the strategic objectives of organization, in order to identify the strategic impacts of current process concerns. The result of this analysis had two significant outcomes:

- It clarified the existing process problems of organization, and helped to specify the goals of Agile adoption initiative - *the to-be process should maximally and effectively address experienced process problems.*
- It clarified the list of organizational strategic objectives, which are negatively impacted by the known process concerns. This issue also helped to further clarify the goals of the Agile adoption initiative – *the to-be process should positively contribute to the organizational objectives which are in need of improvement.*

Having the SG in place, we applied all of the strategic analysis techniques of SAAF, except the trade-off analysis. The outcome of each analysis activity is as follows:

7.7.1 Strategic Contribution Analysis (SCA)

The SCA was conducted for each of the four Agile practices proposed to the R&D unit. For every practice, a copy of the Strategies Graph was used as the basis of analysis. The analysis team in collaboration with selected members of the transition team ran the SCA. The SCA resulted in the specification of strategic objectives that are going to be directly contributed to by any of the candidate Agile practices. For each Agile practice the SCA was carried out by, first, clarifying the definition of the practice; second, extracting the evidence-based contributions; and third, exploring the consensus-based contributions. On average, the SCA of each Agile practice took two hours to be completed. Table 13 shows some statistical information of applying SCA for candidate Agile practices.

TABLE 13: THE NUMBER (#) OF CONTRIBUTIONS OF EACH AGILE PRACTICE TO THE SG OBJECTIVE; THE PERCENTAGE OF EVIDENCE-BASED (EB%) AND CONSENSUS-BASED (CB%) CONTRIBUTIONS; AND SAMPLE STRATEGIC OBJECTIVES WHICH ARE DIRECTLY CONTRIBUTED TO BY EACH AGILE PRACTICE

| Agile Pr. | # | EB% | CB% | Contributed Objectives (sample cases) |
|---------------------|----------|------------|------------|---|
| Scrum Team | 24 | 33% | 67% | Inter Team Communication Collaboration; Faster Problem Solving; Reduced Waiting Times; Improved Supportive Environment; . . . |
| Sprint Planning | 20 | 30% | 70% | Improved Self Organizing; Improved Learning; Improved Inter Org. Communication Collaboration; Facilitated Knowledge Sharing; . . . |
| Daily Scrum Meeting | 16 | 56% | 44% | Facilitated Knowledge Sharing; Inter Organizational Communication Collaboration; Avoid Extra Feature, Doc, handoffs; Self Organization; Reduced Re-Planning; Better Coaching; . . . |
| Short Release | 21 | 42% | 58% | Be Welcoming to Change Requests; Better Understanding of customer / market; Improved Performance; Reduced Defects; . . . |

7.7.2 Propagative Strategic Analysis (PSA)

The next step of the strategic analysis process was to find out the impacts of each practice on higher level strategic objectives of the organization. The contribution tags of strategic objectives, which had received direct contributions, were propagated up the SG. In most cases, the label bag of higher objectives could be resolved automatically, but in some cases, the intervention of process analyst was needed. On average, the PSA of each Agile practice took 30 minutes to be completed.

The SCA and the PSA had the following outcomes:

- *A qualitative measure of each practice* – in terms of its potential positive or negative impacts on organizational strategies. This index was used as an indicator of the strategic impact of adopting the new processes.
- *Descriptive reasoning of each practice* – contribution relations of a CAP to every strategic objective contains the rationale, which explains the reasons of the relation.
- *Shared understanding of each practice* – the overall SAAF framework is intended to foster the participation of organization actors in transitioning to the new process. One of the important result of SCA was the discussions, which was raised while analyzing a practice w.r.t every strategic objective.

7.7.3 Aggregated Strategic Analysis (ASA)

Upon the completion of SCA and PSA for all candidate Agile practices, the ASA was quite straightforward to be conducted. After aggregating the analysis results of individual practices, their overall contribution (in other words the contribution of candidate to-be process) to the strategic objectives of the organization became clarified.

Table 14 presents a part of the table of aggregated analysis developed in the Ericsson case.

TABLE 14: AGGREGATION OF THE CONTRIBUTIONS OF CANDIDATE AGILE PRACTICES (ST: SCRUM TEAM; SP: SPRINT PLANNING; DM: DAILY MEETING; SR: SHORT RELEASE) FOR A SUBSET OF THE STRATEGIC OBJECTIVES

| Strategic Objective | | | Aggregated Contribution | Agile Practice SCA | | | |
|----------------------|-----------------|--|-------------------------|--------------------|----|----|----|
| | | | | ST | SP | DM | SR |
| Develop Good Product | Quality Product | Comply to Standards | Supported | + | | | + |
| | | Design Quality | Declined | | - | | |
| | Right Product | Better Understanding of customer | Supported | + | ++ | + | + |
| | | Close Relationship / Frequent feedback from Customer | Supported | | + | | + |

7.7.4 Strategic Balance Analysis (SBA)

Here we explain the way applied each of the techniques proposed for pre-adoption balance analysis.

7.7.4.1 Strategic Balance Analysis – Balance Improvement

The subject organization had a set of pre-identified strategic objectives that were expected to be improved by the adoption of the new Agile process. These objectives were {Improved Productivity, Reduced Waste, Better Time to Market, Improved Communication / Collaboration}.

It was noticed that the strategic objectives that had been pre-identified for improvement were mostly high-level objectives, which could have been decomposed to, or are contributed

to, by many finer-grained, more specific strategic objectives. This issue became apparent when together with organization members we constructed the Strategies Graph. Questions arose as to which of the high-level strategies were to be further clarified in terms of their constituent (or contributing) lower-level objectives. For instance, the strategic objective 'Reduced Waste' was identified to be decomposable into five other strategic objectives. Having done this, we could identify the exact middle- and low-level strategic objectives which were actually in need of improvement. Finally, we identified 23 strategic objectives, at different levels, which were expected to be improved by the adoption of the new process, resulting in the ToBeImprovedObjs set.

After inspecting the contributions of the fragments of new process over the Strategies Graph, the elements of ToBeImprovingObjs, ToBeDeterioratingObjs, and ToBeConflictingObjs were identified. In the experiment, the ToBeImprovedObjs had four elements in common with the ToBeConflictingObjs, but no element in common with the ToBeDeterioratingObjs. However, the common set of ToBeImprovedObjs and ToBeConflictingObjs were middle- and high-level strategic objectives, each was contributed to by a number of lower-level strategic objectives.

Another result of this analysis was the following observation: in most cases the lower-level strategic objectives would belong either to the ToBeDeterioratingObjs or ToBeImprovingObjs, or they do not receive any contribution from the new process; however, middle- and high-level strategic objectives often receive conflicting contributions from the new process and belong to the ToBeConflictingObjs (which is quite reasonable because the received contribution of a high-level objective is the aggregation of contributions that its lower-level objectives received).

7.7.4.2 Strategic Balance Analysis – Balance Preservation

Among the strategic objectives that were not in the focus of improvement, two were identified with $CW < 0$; one of them belonged to the ToBeDeterioratingObjs and the other one to the ToBeConflictingObjs. The deteriorating one was then identified to be supported by some of the already in-place development techniques, which were going to remain unchanged after the adoption of the new process. Thus, the negative impact of the new process to this objective was decided to be tolerable. However, for the conflicting strategic objective, the trade-off analysis failed to find any already in-place technique, or any other fragment of new process to

make it up. This issue led the analysis team to decide on a new supportive action, to preserve the balance of that particular strategic objective.

It should be mentioned that the new process had only four fragments. These fragments were specifically chosen by the process design team to address some of the issues of the current process. If the to-be process was more complex, it could be expected that a greater number of strategic objectives would have negative CWs. The reason for this assumption is that at least from the business perspective, the cost of incorporating a complex set of changes (which radically alters the current process) is more than the cost of modifying current process with a limited set of fragments. Therefore, for complex process adoption initiatives, strategic objectives which belong to the Business category are more prone to be side-effects of the initiative.

7.7.4.3 Strategic Balance Analysis – Balance Preservation across Categories

After calculating the total weight of positive and negative contributions (TWPC and TWNC) the total weight of contribution (TWC) of all categories turned out to be positive. Indeed, none of the categories received a substantially bad TWNC, which indicated that the adoption of the new process would not cause the overall deterioration of one category of objectives as a side-effect of improving some other categories.

As mentioned before, this analysis was carried out while another existing initiative of Agile adoption was already in progress. In our observation, this fact might have indirectly influenced the way stakeholders expressed strategic objectives, predisposing them towards Agile values. For this reason, the candidate Agile practices were mostly analyzed to have positive impacts over strategic objectives, with few negative impacts. In general, this issue – i.e., incomplete or biased construction of the Strategies Graph – can be considered as a threat to the validity of analysis results.

7.7.4.4 Strategic Balance Analysis – Homogenous Contributions across Categories

In this analysis exercise, strategic objectives were grouped into three categories: User Orientation, Business Value, and Internal Process, respectively containing 16, 15, and 20 strategic objectives. Table 15 shows the distribution of the number and percentage of strategic objectives with different statuses in each of the three categories. For instance, it shows that

80% (16 out of 20) of the strategic objectives of the Internal Process category would be supported by the fragments of the new process, with 10% conflicted, 10% unaddressed, and none declined.

TABLE 15: CONTRIBUTION DISTRIBUTION TABLES (U: UNADDRESSED, C: CONFLICTED, S: SUPPORTED, D: DECLINED)

| User Orientation | | | | | Business Value | | | | | Internal Process | | | | |
|------------------|---|---|----|----|----------------|---|----|----|----|------------------|----|----|----|----|
| | | | | 16 | | | | | 15 | | | | | 20 |
| | U | C | S | D | | U | C | S | D | | U | C | S | D |
| # | 0 | 0 | 15 | 1 | # | 0 | 3 | 13 | 0 | # | 2 | 2 | 16 | 0 |
| % | 0 | 0 | 93 | 7 | % | 0 | 20 | 80 | 0 | % | 10 | 10 | 80 | 0 |

The variances of numbers and percentages of different statuses across categories were then calculated. The results showed that the new process makes a homogenous contribution to the Supported strategic objectives across different categories, as the variances of the percentages and numbers were quite low. The contributions to Declined strategic objectives also had low variances.

7.8 Evaluation of Strategic Analysis Results

The objective of this evaluation was to investigate the extent of the correctness of framework's analysis reports on strategic analysis of Agile practices. For instance, if the framework had reported that the enactment of Agile practice *PI* would make *strong positive* contribution to the strategic object *SI*, the evaluation process was to investigate whether the anticipated type of contribution was actually observed after the enactment of the new process. Since SAAF is designed to be used at the early stages of transitioning to Agile (pre-adoption stage), the evaluation of its results would have to be deferred until the transition is completed, and the new process is stabilized within the target organization. Thus, we waited for about a year after the application of SAAF in our collaborating organization (Ericsson), and then investigated the degree of the accuracy of framework results. It should be mentioned that the results *Strategic Balance Analysis* procedure were not evaluated in this process.

7.8.1 Evaluation Process

As mentioned before the SAAF was applied during the process of transitioning to an Agile method in of the R&D units of the Ericsson. Due to the nature of the framework, it was applied at the earlier stages of this transition, and helped the team to anticipate what strategic impacts this transition would make on their organizational goals.

When the team completed the transition to Agile, and stabilized the new process in their organization, it was possible to evaluate the accuracy of framework result. We conducted a survey, asking organizational managers about the degree of the accuracy of SAAF analysis results.

7.8.2 Evaluation Result

As explained in Appendix 1 (SAAF Evaluation Questionnaire), participants were asked to express their degree of agreement with the analysis results of the SAAF, in the scale of 1 to 5, where 5 was the Strongly Agree and 1 was the Strongly disagree. The following subsections explain the evaluation result of the four Agile practices which were under study.

TABLE 16: LIST OF ANALYZED AGILE PRACTICES, AND TOTAL NUMBER OF CONTRIBUTIONS THAT EACH AGILE PRACTICE WOULD MAKE TO ORGANIZATIONAL OBJECTIVES

| Agile Practice | # of Contributions to Org. Objectives |
|-------------------------------|--|
| Scrum Team Structure | 25 |
| Sprint Planning | 21 |
| Daily Scrum Meeting | 16 |
| Potentially Shippable Product | 22 |
| Total = 84 | |

Table 17 shows Number of contribution relations from each Agile practice, which received certain agreement degree. For instance, it shows that for the Agile practice “Scrum Team Structure”, participants were “Strongly Agree” with 18 (out of 25) of the originally anticipated contribution relations, “Agree” with 4, “Neither Agree nor Disagree” with 2, and “Strongly Disagree” with 1 of them. This implies that for that Agile practice the framework analysis results were over 80% in accordance with what was actually observed in reality.

TABLE 17: NUMBER OF CONTRIBUTION RELATIONS FROM EACH AGILE PRACTICE WHICH RECEIVED CERTAIN AGREEMENT DEGREE

| | Strongly Agree | Agree | Neither Agree nor Disagree | Nor Disagree | Strongly Disagree |
|-------------------------------|-----------------------|--------------|-----------------------------------|---------------------|--------------------------|
| Scrum Team Structure | 18 | 4 | 2 | 0 | 1 |
| Sprint Planning | 13 | 0 | 0 | 5 | 3 |
| Daily Scrum Meeting | 9 | 1 | 2 | 0 | 4 |
| Potentially Shippable Product | 20 | 0 | 2 | 0 | 0 |
| Total # | 60 | 5 | 6 | 5 | 8 |
| Total % | 71.43 | 5.95 | 7.14 | 5.95 | 9.52 |

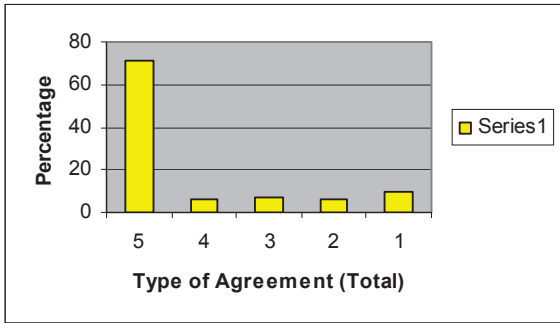
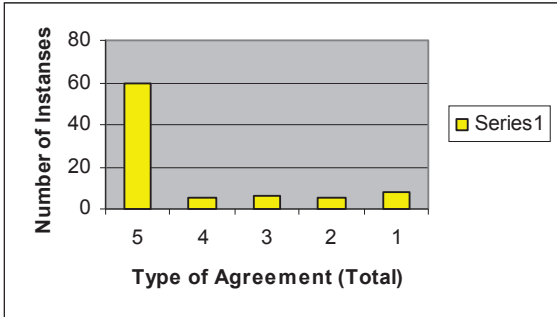


FIGURE 30: OVERALL DISTRIBUTION OF THE NUMBER AND THE PERCENTAGE OF AGREEMENT TYPES

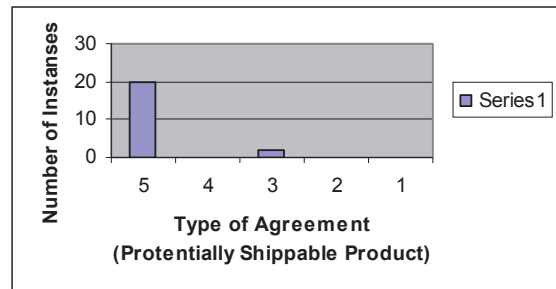
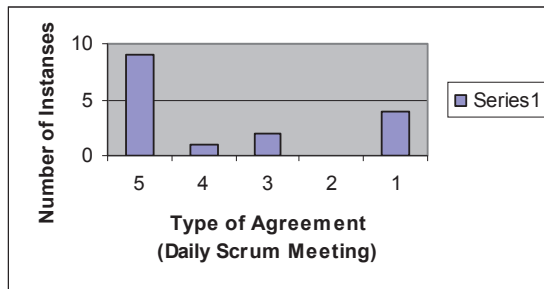
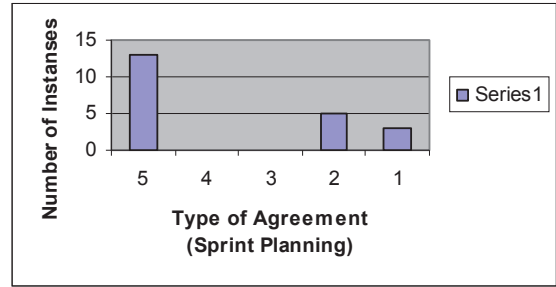
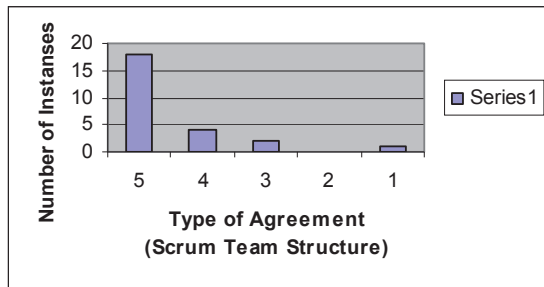


FIGURE 31: DISTRIBUTION OF THE NUMBER AND THE PERCENTAGE OF AGREEMENT TYPES FOR EACH CANDIDATE AGILE PRACTICE

TABLE 18: DISTRIBUTION OF AGREEMENTS WITH THE CONTRIBUTION RELATIONS TO ORGANIZATIONAL OBJECTIVES, WHICH WERE UNDER THE CATEGORY OF INCREASED PRODUCTIVITY OF INDIVIDUALS

| | Strongly Agree | Agree | Neither Agree Nor Disagree | Disagree | Strongly Disagree |
|-------------------------------|-----------------------|--------------|-----------------------------------|-----------------|--------------------------|
| Scrum Team Structure | 5 | 4 | 1 | | |
| Sprint Planning | 4 | | | 1 | 2 |
| Daily Scrum Meeting | 3 | 1 | 1 | | 2 |
| Potentially Shippable Product | 4 | | 1 | | |
| Total # | 16 | 5 | 3 | 1 | 4 |
| Percentage % | 55.17 | 17.24 | 10.34 | 3.45 | 13.79 |

TABLE 19: DISTRIBUTION OF AGREEMENTS WITH THE CONTRIBUTION RELATIONS TO ORGANIZATIONAL OBJECTIVES, WHICH WERE UNDER THE CATEGORY OF EFFICIENT COMMUNICATION / COLLABORATION AND REDUCED DEVELOPMENT COST

| | Strongly Agree | Agree | Neither Agree Nor Disagree | Disagree | Strongly Disagree |
|-------------------------------|-----------------------|--------------|-----------------------------------|-----------------|--------------------------|
| Scrum Team Structure | 8 | | | | 1 |
| Sprint Planning | 5 | | | | |
| Daily Scrum Meeting | 4 | | 1 | | 2 |
| Potentially Shippable Product | 7 | | | | |
| Total # | 24 | 0 | 1 | 0 | 3 |
| Percentage % | 85.71 | 0.00 | 3.57 | 0.00 | 10.71 |

TABLE 20: DISTRIBUTION OF AGREEMENTS WITH THE CONTRIBUTION RELATIONS TO ORGANIZATIONAL OBJECTIVES, WHICH WERE UNDER THE CATEGORY OF TIME TO MARKET, QUALITY PRODUCT, AND CUSTOMER SATISFACTION

| | Strongly Agree | Agree | Neither Agree Nor Disagree | Disagree | Strongly Disagree |
|-------------------------------|-----------------------|--------------|-----------------------------------|-----------------|--------------------------|
| Scrum Team Structure | 5 | | 1 | | |
| Sprint Planning | 4 | | | 4 | 1 |
| Daily Scrum Meeting | 2 | | | | |
| Potentially Shippable Product | 9 | | 1 | | |
| Total # | 20 | 0 | 2 | 4 | 1 |
| Percentage % | 74.07 | 0.00 | 7.41 | 14.81 | 3.70 |

7.8.3 Discussion on Evaluation Results

If members of the evaluation team were the same as the members of contribution analysis team, it is highly possible that they repeat their original opinions about the contribution relations. In fact, this was not the case, and in Ericsson experience, there was substantial difference between the members of analysis and evaluation team, though there were some members who participated at both activities. Therefore, the above mentioned threat is not considered as a serious one.

Some participants might have been biased about the analysis results, even at the time analysis, and that bias was extended to the evaluation results. One possible scenario is that a participant had disagreed with some particular contribution relation (at the time of contribution analysis), and his opinion was masked by others. So, at the time of evaluation, he expressed his original opinion about the contribution relation, rather than the actual observation. This can be counted as a threat to the validity of Evaluation results, though its scope should not be considerable.

7.9 Experience with Strategic Actor Analysis

In the Ericsson experience we used Itemized Strategic Dependency Diagrams for facilitating the extraction of as-is process concerns. Once the concerns were identified, the Strategies Graph was used to analyze the impacts of each concern.

7.9.1 Using Itemized Dependency Diagrams for Extracting Process Concerns

The activity of extracting as-is process concern began by identifying different roles in the R&D unit, and selecting a number of representatives from each role, depending on the number of people in each role. In total, we identified the following roles in the target R&D unit:

1. Project Manager
2. Solution Manager
3. System
4. Designer
5. Developer
6. Test Coordinator
7. Tester

The next step in extracting process concerns was to interview the representatives of each role. In fact, we conducted two rounds of interview with the members of transition team, individually or in groups of two. For instance, the project manager and chief test coordinator were interviewed individually, but representatives of the test team were interviewed in groups of two.

Before starting the first round of interviews, initial drafts of Functional ISDs were developed. The basic assumption was that there should be some sort of dependency among all of the organization roles. Thus, the initial FISDs were developed in a simplified way, which were just showing the two-by-two dependency relations of various organization roles (i.e., each of the developed FISDs were containing only the hypothetical dependency relations of two roles). Thus, having six organization roles, 21 FISD diagrams were developed, showing the mutual dependencies of all organizational roles. Figure 32 shows one of the initial FISDs, developed to highlight the dependency relations of Solution Manager and System Manager.

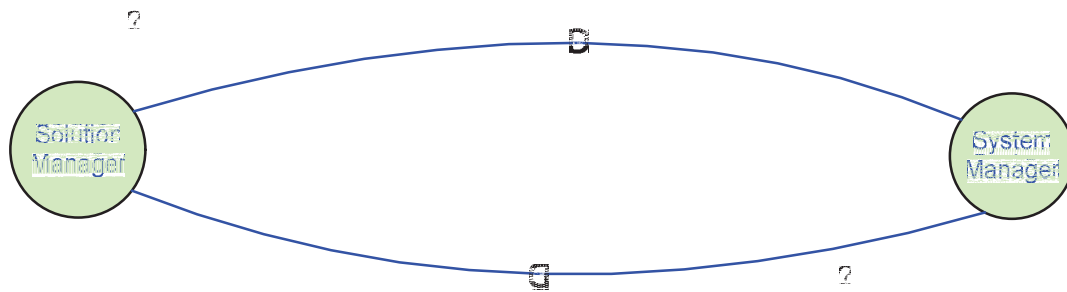


FIGURE 32: INITIAL FISD OF SOLUTION AND SYSTEM MANAHER ROLES

The initial FISDs were incomplete, since at that stage of the analysis, the dependency objects were unknown. Thus, question marks were places instead of actual Functional Dependencies. Having the initial FISDs developed, the first round of interviews was started with the following agenda:

1. Introducing the whole project, which was about transitioning to Agile
2. Introducing (briefly) the overall SAAF
3. Describing the approach in extracting process concerns
4. Training the interviewee in FISDs
5. For each initial FISDs that contained the interviewee's role
 - a. Showing the FISD and explaining the idea behind it
 - b. Asking to express functional dependencies to and from the other role
 - c. Asking to express some of the concerns in mutual dependency with the other role

After the first round of interviews, a very good understanding of functional dependencies among various organization roles was gained. Indeed, some of the process concerns which were related to such dependency relations were extracted. Due to time constraints, during the first round of interviews, interviewees were not asked to complete the FISDs, instead, FISDs were just being used to guide the interview process, and keep the focus of the meeting on the subject matter.

After each interview, the extracted information was added to the electronic version of FISDs, using Microsoft Visio. By this time, the FISDs were containing a short description of all dependency relations in the target R&D unit. Figure 33 shows a sample FISD, developed after the first round of interviews.

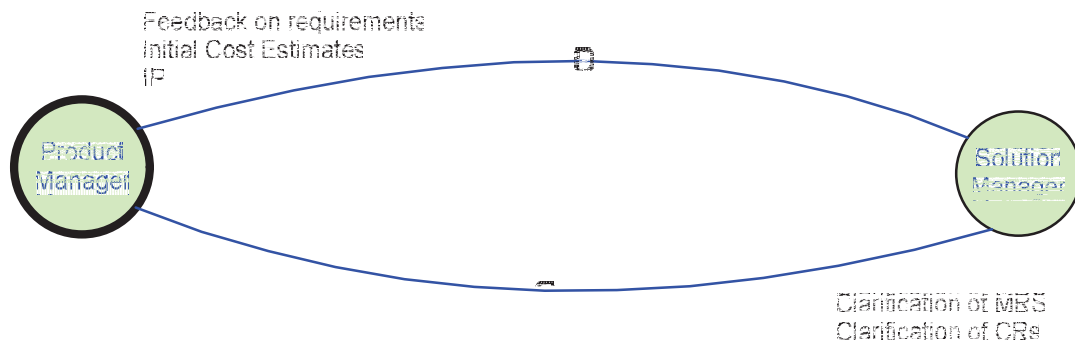


FIGURE 33: SAMPLE FISD, DEVELOPED AFTER THE FIRST ROUND OF INTERVIEWS

Gaining an understanding of the functional dependencies among various organization roles, prepared the situation for running the second round of interviews. In this round, similar to the previous one, interviewees were met individually or in groups of two. The following agenda was used leading each session:

1. Showing the printed version of the FISD developed after the first meeting
2. Asking for completing any missing functional dependency
3. Asking for expressing any quality attribute, related to any of the functional dependencies
4. Asking to evaluate each of the functional dependencies and their related quality attributes
 - a. For functional dependencies, a two value metric was used: {Fulfilled, Unfulfilled}

- b. For Quality attributes, a four value metric was used {Fulfilled, Partially Fulfilled, Partially Denied, Fully Denied}. A graphical notation similar to *i** evaluation marks was used to facilitate this process.

In fact, during the second round of interviews, the FISDs were elaborated to become QISDs, expressing the quality attributes of functional dependencies. By putting the focus of meeting on functional dependencies which were unfulfilled, and quality attributes which were somehow denied, ISDs helped us to extract a good number of issues, which were perceived as process concerns by various organization roles. For instance, the QISD shown in Figure 34 shows that Designers had been expecting the System Manager to supply NRS Document (a kind of specification document), in such a way to be Sufficiently Detailed, Regularly Updated, Technical, Clear, and understandable. However, many of these quality attributes were either denied or poorly met.

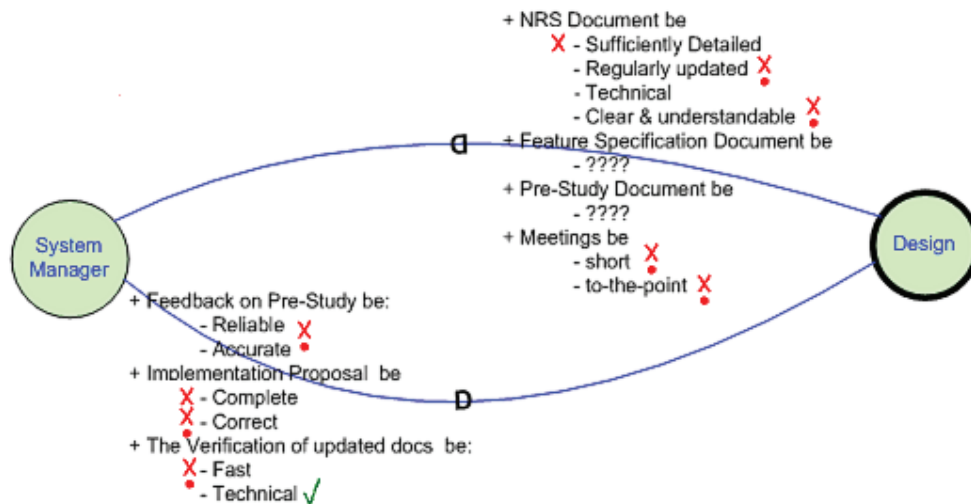


FIGURE 34: SAMPLE EVALUATED QISD

After the second round of interviewing, the electronic version of documents were updated. At the end of this stage, over 100 issues were extracted as perceived process concerns of the R&D unit.

7.9.2 Analyzing the Extracted Process Concerns

The analysis of extracted process concerns happened in three stages: 1 – Classification, 2 – Root-Cause Analysis, and 3 – Strategic Analysis. The following subsections explain each stage in detail.

7.9.2.1 *Classifying Process Concerns*

The initial set of process concerns extracted through the ISDs was unorganized, and did not have a unified terminology. The initial process concerns went through a process of classification; and unification of the terminology. As a result, five categories of concerns were identified, which would classify all of the process concerns. These categories of concern are as follows:

- Requirements Problems
- Documentation Problems
- Communication / Collaboration Problems
- Project Planning and Resource Management problems

The following tables show a selected number of process concerns belonging to each category.

TABLE 21: SAMPLE PROCESS CONCERNS UNDER THE CATEGORY OF REQUIREMENTS PROBLEMS

Requirements Problems

- No direct contact between designers /customer (for customer part)
- Performance metrics of system test are not well-defined
- Development by assumption
 - Development of extra (sometimes complex) functions
 - Not development of needed functions
 - Wrong development of functions behaviors
- Requirements change propagation
 - Requirements documents change and that makes strong change on the test document.
 - Requirements changes but the relevant documents are not kept updated (e.g. IP, Pre-Study doc, Solution Spec at LMI level)

TABLE 22: SAMPLE PROCESS CONCERNS UNDER THE CATEGORY OF DOCUMENTATION PROBLEMS

Documentation Problems

- Extra documents
 - Some designers believed that in many cases the IP is not needed, as it will change, not be kept updated. Besides, in many of the technical committee meetings we receive no useful comment. We are the ones who write the IP, and implement the features.
 - In some cases code can be developed even directly without detailed NRS, *Well, some others had completely contradictory idea!!*
 - Unnecessary or Inefficient reviews or approvals
- Ineffective documents
 - Some designers were complaining that the level of *Technical Details* in NRS document is too low.
 - Documents are not written in your first language, so they are subject to misrepresentation and misinterpretation (According to one designer!)
 - Issues about content and format of documents (e.g. NRS is not always to-the-point, and does not contain enough technical Detail)

TABLE 23: SAMPLE PROCESS CONCERNS UNDER THE CATEGORY OF COMMUNICATION /
COLLABORATION PROBLEMS

Communication / Collaboration Problems

- Personal knowledge of software architecture
- Localized knowledge of subsystem design
- Not all people know the big picture but just the piece of process s/he is involved with, without clear idea of what happened after them.
- Misperceptions of testers about the code to test
- Hard possibility of common code ownership
- Lack of a shared document that about system architecture
- *Incomplete views, different interpretations, assumptions, delay in accessing the needed knowledge . . .*
- CC issues between designers and testers
 - Frequent interruptions on designers
 - *Well, different ideas !!*
 - Missed test cases, for some functionalities
 - New test cases, for never known functionalities

- Lots of meetings between designers and testers to clarify the expected and actual system behavior

TABLE 24: SAMPLE PROCESS CONCERNS UNDER THE CATEGORY OF PROJECT PLANNING AND RESOURCE MANAGEMENT PROBLEMS

Project Planning and Resource Management Problems

- Estimation-based resource allocation
- Estimations are not always reliable
- Designers suffer from overtime work. Few time to learn.
- Overcommitted subsystem responsible(s) create inefficiencies due to “changeover” times
- Planning for multiple development iterations
- Possibility of requirements change, and change of plan
- Initial documents cannot be reliable enough for planning big ahead
- Test plans do not cover all of the functionalities of a new build (developed by designers)
- Test Plans, in some cases, are not in accordance with the released code

- Project manager and technical coordinator are not always aligned with the prioritization of design tasks

Furthermore, the process concerns of each category were divided into a number of *areas of concern*. For instance the following areas of concern were identified for the category of Requirements Problems:

- Problematic Requirements Specification
- Development by assumption
- Cost of Requirements Change
- Requirements Analysis
- Validity of Requirements

7.9.2.2 Root-Cause Analysis of Process Concerns

In order to further organize the extracted process concerns, a root-cause analysis was conducted among them, to identify which process concerns can be considered as root or cause of some other. Since all of the process concerns were a kind of “cause” for some negative contribution to organizational strategic objects, process concerns were classified to *Root* and *Intermediate* Causes, where intermediate ones were considered as the result of root causes. For example, the following table shows the result of root-cause analysis among the process concerns of Category: Requirements Problems, and Area: Cost of Requirements Change:

TABLE 25: SAMPLE ROOT-CAUSE ANALYSIS

1.3 Area of Concern: Cost of Requirements Change

- **(Root Cause)** Change of requirements

- **(Intermediate Cause)** Change of test plans
- **(Intermediate Cause)** Rework in system design and implementation
- **(Intermediate Cause)** Updating the relevant documents, e.g. IP, Pre-Study doc, Solution Spec upon req. change. *These updates are not usually happening!!!*

7.9.2.3 Strategic Analysis of Process Concerns

In order to clarify the strategic impacts of the identified process concerns, concerns of each category were separately exposing the strategies graph, and their negative impacts were evaluated. For instance, Figure 35 shows the model-driven strategic analysis of process concerns of the category “Documentation Problems” over a portion of the Strategies Graph. Color coding is used to distinguish various areas of process concerns.

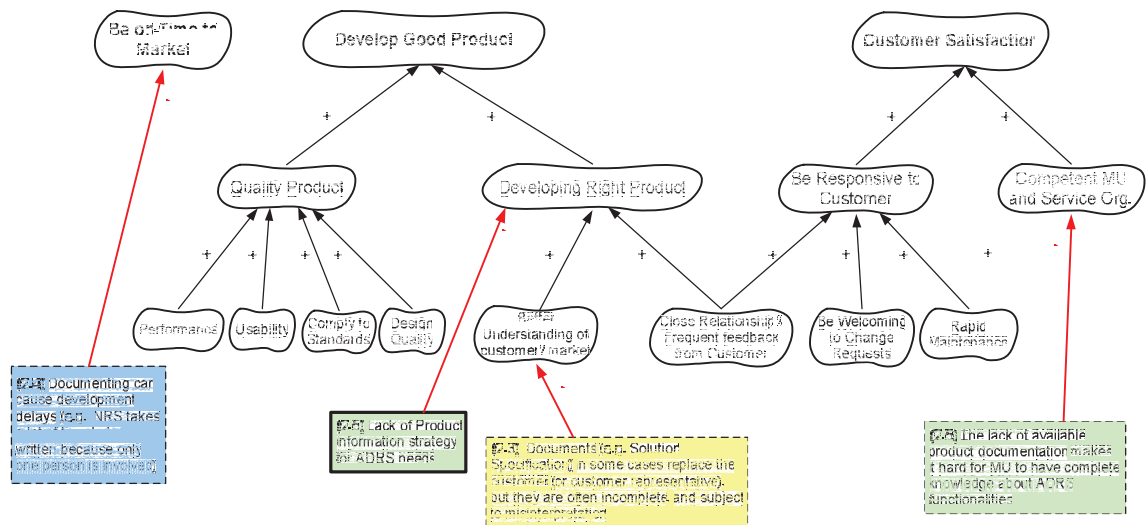


FIGURE 35: STRATEGIC ANALYSIS OF PROCESS CONCERNS OF CATEGORY "DOCUMENTATION PROBLEMS"

After conducting the Strategic Analysis of process concerns, the analysis results were converted to tabular format, showing how many negative contributions is being made to any strategic objective. These tables were then complemented by the result of Contribution Analysis of Candidate Agile Practices, showing the positive or negative contributions made to any strategic objective. Table 26 shows one of such tables.

TABLE 26: THE IMPACT TABLE OF SUB-GOALS OF "EFFICIENT DEVELOPMENT PROCESS"

| | | Reduced Development Cost | | | | | | | | | |
|--------------------|---|--------------------------|-------------------------------|------------------------|------------------|-----------------|----------|------------------------------------|-----------------------|----------|--|
| | | Reduced Maintenance cost | | Improved Output | | | | Avoid Wastes | | | |
| | | | Facilitated Knowledge Sharing | Faster Problem Solving | More LOC / Month | Reduced Defects | | Avoid Extra Feature, Doc, handoffs | Reduced Waiting Times | Reuse | |
| | | | | | | | | | | | |
| Candidate Practice | Scrum Team | | + | + | + | | | + | ++ | | |
| | Sprint Planning | | + | | | + | | + | | | |
| | Daily Meeting | | + | | | + | + | + | + | | |
| | Short Release | | | | | + | | ++ | | | |
| Concerns | Total # of concerns impacting Objective | | 5 | 3 | 5 | 6 | 4 | 8 | 4 | 3 | |
| | # of Req. Eng. Concerns impacting SGoals | | | 1 | 2 | 1 | | 4 | 1 | 1 | |
| | # of Documenting Concerns impacting SGoals | | 1 | 1 | 2 | 2 | | 2 | 2 | 2 | |
| | # of Commu. Collab. Concerns impacting SGoals | | 3 | 1 | 1 | 2 | 2 | 2 | 1 | | |
| | # of Proj. Plan. Rsc. Mang. Concerns impacting SGoals | | | | | | 1 | 2 | | | |

Impact tables were used to extract the following knowledge:

- Strategic objectives which are negatively impacted by process concerns, but supported by Candidate Agile Practices as well. For instance, Table 26 shows that the strategic objective “Avoid Extra Feature, Doc, Handoffs” is being negatively impacted by 8 process concerns; however, the objective is being supported by all four Candidate Agile Practices.
- Strategic objectives which are negatively impacted by process concerns, but are not well supported by Candidate Agile Practices. For instance, Table 26 shows that the objective “Reuse” is being negatively affected by 3 process concerns, but none of the candidate practices would support it. Or the strategic objective “More Line of Code (LOC) / Month”, is being affected by 5 process concerns, and supported by only one candidate practice.

The strategic analysis of process concerns were complemented by the analysis, explained in the next subsection.

7.9.3 Agile Practices Capability Analysis

The final stage of concerns analysis was to investigate the capabilities and pitfalls of the Candidate Agile Practices in addressing any of the identified process concerns. Having the description of every Candidate Agile Practice, the analysis team went through each of the identified process concerns to see if it can be addressed by any of the candidate Agile practices. Table 27 shows a sample of such analysis, for the ways that the Agile practice “Daily Scrum Meeting” can address a subset of the “Requirements Engineering” process concerns.

TABLE 27: LIST OF CONCERNS ADDRESSABLE BY DAILY SCRUM MEETING (DSM), AND THE WAY DSM TREAT THEM

| R&D UNIT Concerns Addressable by DSM | The way DSM treats the R&D UNIT concerns |
|---|--|
| (R.C.) Incomplete specification of system behavior | Requirements problems can be raised at DSMs, and avoid development by assumption |
| (I.C.) Frequent meetings for requirements clarification | DSM frequency can be adjusted based on organization / project situation |
| (I.C.) Development of extra (sometimes complex) functions | Development decisions gets confirmation at DSMs |
| (I.C.) Not development of needed functions | Development decisions gets confirmation at DSMs |
| (I.C.) Wrong development of functions behaviors | Development decisions gets confirmation at DSMs |
| (I.C.) Change of test plans | Change of requirements can be discussed at DSMs, and related test plans get modified |
| (I.C.) Rework in system design and implementation | Development decisions gets confirmation at DSMs and this will help team to reduce rework |

When this analysis was completed for all candidate Agile practices (over all of the identified process concerns), the analysis results were aggregated to identify the overall coverage of

candidate Agile practices, in addressing as-is process concerns. Table 28, shows a portion of such tables.

TABLE 28: PROJECT PLANNING & RESOURCE MANAGEMENT CONCERNS & CANDIDATE AGILE PRACTICES (STS: SCRUM TEAM STRUCTURE; SP: SPRINT PLANNING; DSM: DAILY SCRUM MEETING; SR: SHORT RELEASE)

| | STS | SP | DSM | SR | NON |
|---|-----|----|-----|----|-----|
| 4.1 Area of Concern: Planning based on early estimations | | | | | |
| (R.C.) Estimations are not always reliable | | ✓ | | | |
| (R.C.) Initial documents cannot be reliable enough for planning big ahead | | ✓ | ✓ | | |
| (I.C.) Designers suffer from overtime work. Few time to learn. | | ✓ | | | |
| (I.C.) Overcommitted subsystem responsible(s) create inefficiencies due to "changeover" times | ✓ | | | | |
| 4.2 Area of Concern: Planning for multiple development iterations | | | | | |
| (R.C.) Big planning ahead | | ✓ | | ✓ | |
| (I.C.) Possibility of requirements change, and change of plan | | ✓ | ✓ | | |
| 4.3 Area of Concern: Test Planning ahead | | | | | |
| (R.C.) Test planning at early phases of work, where requirements are still subject to change | | ✓ | | | |
| (I.C.) Test instructions, in some cases, are not in accordance with the released code: (1) implemented functionalities that were not tested, or (2) New test cases, for never known functionalities; resulting in rework for designers / testers. | ✓ | ✓ | ✓ | | |
| 4.4 Area of Concern: Task Prioritization | | | | | |
| (R.C.) No participatory approach for prioritization of development tasks | ✓ | ✓ | | | |
| (R.C.) Project manager and technical coordinator are not always aligned with the prioritization of design tasks | | ✓ | | | |
| (I.C.) Designers sometimes focus on low priority tasks, mainly coming from test or maintenance teams, rather than focusing on core functions that have higher priority. | | | ✓ | | |

Out of such tables the following knowledge can be extracted:

- Which of the as-is process concerns will most probably remain even after the introduction of the new Agile method – those which are not addressed by any of the candidate Agile practices.
- Which of the as-is process concerns should be resolved after the introduction of the new method – those which are being well addressed by candidate practices.

After completing this analysis the capability of selected Agile practices in addressing current process concerns was clarified. Table 29 shows the statistical summary this analysis. The provision of this kind of detailed analysis results to R&D project managers, helped them to anticipate which set of process concerns can be resolved by the introduction of new Agile method; and which subset will remain unaddressed or poorly addressed.

TABLE 29: TOTAL NUMBER OF CONCERNS AT EACH CATEGORY OF CONCERN, AND THE COVERAGE OF CANDIDATE AGILE PRACTICES IN ADDRESSING CONCERNS OF EACH CATEGORY (RE: REQUIREMENTS ENGINEERING PROBLEMS; DP: DOCUMENTING PROBLEMS; CC: COMMUNICATION / COLLABORATION PROBLEMS; PP&PM: PROJECT PLANNING AND RESOURCE MANAGEMENT PROBLEMS)

| | RE | DP | CC | PP& RM | Total |
|--|----|----|----|--------|-------|
| Total Number of Concerns | 18 | 22 | 27 | 14 | 81 |
| Total Number of Root Concerns | 10 | 6 | 11 | 9 | 36 |
| Total Number of Intermediate Concerns | 8 | 16 | 16 | 5 | 45 |
| Number of concerns not Addressed by any of APs | 5 | 14 | 11 | 2 | 32 |
| Number of concerns addressed by one AP | 3 | 6 | 11 | 6 | 26 |
| Number of concerns addressed by two APs | 6 | 2 | 4 | 5 | 17 |
| Number of concerns addressed by three APs | 4 | 0 | 1 | 1 | 6 |
| Number of concerns addressed by four APs | 0 | 0 | 0 | 0 | 0 |

7.10 Final Results of Applying SAAF in Ericsson project

After applying the full cycle of the SAAF in Ericsson project, unit managers were presented the complete analysis results. One of the items which at first sight appear to be omitted in SAAF is to generate a final decision whether to adopt an Agile practice or not. In fact, SAAF has intentionally avoided such proposals, as it is a framework aimed at analyzing the potential outcomes of transitioning to Agile, not to strictly generate a YES/NO answer on candidate practices. In the Ericsson project, what happened after presenting the analysis results to the managers, they found the following information quite helpful:

- List of built-in objectives of each Agile practice
- List of strategic objectives being affected by the candidate Agile practices
- List of as-is process concerns of the organization.
- List of strategic objectives being affected by the as-is process concerns.
- Capabilities and shortcomings of new Agile method in addressing the as-is process concerns.
- Capabilities and shortcomings of the new Agile method in supporting key organizational strategic objectives.
- The pre-conditions needed for making a transition to the new Agile method.

After reviewing the SAAF results, unit managers came to this conclusion that the currently selected set of Agile practices are not adequate for addressing the current problems. In fact, they came to this conclusion that their R&D unit is suffering from some problems (as-is process concerns), which will most probably remain unresolved even after enduring all costs of transitioning to Agile. However, regarding the potentially positive contributions of new method to their key strategic objectives, the choice of transitioning to Agile did not get off the table. What happened is that the organization temporarily ceased the process of transitioning to Agile; conducted some training sessions on Agile (specially the Agile values, rather than practices); and then built up an action plan for removing its as-is process concerns (which were not addressable by their Agile method).

Chapter 8. Conclusion and Future Work

While operational managers deal with the question of “Are we doing things right?”, strategic managers are mainly concerned with answering “Are we doing the right thing?” (Porter, 1996). In the context of method adoption, choosing the right process is as important as its proper implementation. This includes the strategic alignment of the new process with the enterprise-wide, as well as team- and even individual-level strategies and objectives.

Despite the widespread promotion of a number of Agile processes, for many companies becoming Agile means incorporating a small number of Agile practices in their existing development methods. Empirical studies show that among software companies that claim to be Agile, only a few are actually following the complete process of any Agile method as originally prescribed (Aranda *et al.*, 2007). In most cases, software companies would build their Agile methods through ad-hoc assembly of Agile practices, relying mostly on the experience of project gurus or some external consultants. It has been widely reported that Agile practices do not always fulfill their advertised promises, as their ultimate results depend to the situational characteristics of adopting organization and project. Therefore, a safe transition to Agile methods (or in a broader perspective, to any new method) is one that happens with respect to the situational characteristics and strategic objectives of organizations.

When an organization decides to incorporate a set of Agile practices into its development process, it needs to have clear statements for the following questions:

- What are the motivations of changing the current process?
- Is the selected set of Agile practices the right choice for our organization?

The primary motivation for process improvement initiatives is often to resolve the insufficiencies of the current development process, with the aim of reaching strategic goals such as better performance, productivity, time-to-market, customer satisfaction, etc. In other words, most process improvement initiatives are supposed to target the current issues of an

organization, which are somehow related to the development process. Thus, having a clear understanding about the current issues, and the ways they are affecting the strategic goals of organization is an important step which should be taken before adopting a new process.

Having a clear understanding about the needs for process improvement, software organizations that are willing to adopt Agile are interested to know: (1) To what extent the selected set of Agile practices will address their current issues; (2) what benefits can be expected to be achieved when incorporating any of the candidate Agile practice; (3) How might any of the candidate practices impact the strategic goals of organization; and (4) what conditions are expected for the success of candidate practices. Answers to these questions help a project/process manager to decide about the inadequacies and insufficiencies of the candidate Agile practices.

Considering the increasing trend of transitioning to Agile methods, this thesis introduced a framework for evaluating an Agile method, prior to its enactment in a software development company. The proposed framework (SAAF) helps organization managers to anticipate the potential impacts of transitioning to Agile on key organizational strategic objectives. Indeed, the framework assists organizations to have a thorough understanding of their as-is process concerns, and anticipate whether these concerns will be resolved after the adoption of new method.

SAAF has two unique characteristics:

1. Being built on top of the empirical knowledge of Agile methods
2. Taking a model-driven approach for strategic analysis

The prevalence of Agile methods has provided a suitable test-bed for software researchers, to run empirical studies, and analyze different aspects of Agile methods in real trials. Every year, a considerable number of empirical studies emerge, each addressing specific concerns of deploying Agile methods in various projects/organizations. These studies form a large body of knowledge about various fragments (constituent pieces) of Agile methods. For instance, a simple title-based search in just the IEEE Explorer for "Pair Programming" returns over 50 results, mostly reporting on empirical studies. As one of its key components, SAAF is having a *Repository of Agile Practices*, which provides an structured summary of the results of many empirical studies on various Agile practices. This repository has been build out of a systematic

literature review of empirical studies, published on different Agile methods. The repository is available online at www.ProcessExperiecen.org.

SAAF takes advantage of the i^* modeling framework for its strategic analysis procedures. The framework defines a variant of base i^* modeling notation, for visualizing the strategic model of an organization. The developed model, which is called Strategies Graph, will then be used as the basis of a number of analysis procedures, aimed at exploring the strategic advantages, or disadvantage, of candidate Agile method. The same graph will be also used as part of analyzing the as-is process concerns of the organization.

Taking a model-driven approach for the analysis of Agile methods, such as what SAAF does, can raise this confusion that it is in conflict with nature of Agile methods. On the other hands, the method analysis procedures can take some time, and hinder the speed of becoming Agile. In response to these questions, we should mention two items: first, Agile methods are not against modeling, as they advocate enough (rather than extensive) amount of modeling in requirements engineering. Therefore, using models in an Agile context is not forbidden, or in conflict with the nature of Agile. Second, though Agile – from one perspective – is about rapid software development, the transitioning to Agile does not need to be instantaneous. Spending a few more days before making the transition to a new method to evaluate it from different perspectives can save an organization considerable amounts needed to overcome the negative impacts of a wrongly selected method.

It is widely reported that one of the main barriers in transitioning to Agile is the hesitance of managers in the expected outcomes of new method. SAAF addresses this issue by investigating the impacts of the new method over the strategic concerns of organization at different levels. In this regard, the *Propagative Strategic Analysis* is proposed to trace the impacts of the new method over the higher-level strategic goals, which are of interest to higher-level managers. *Aggregated* and *Balance Analyses* are proposed to anticipate the impact of a set of Agile practices when they get applied all together. The Ericsson experience showed that these kinds of strategic analyses make the process of transitioning to Agile more predictable. Indeed, the analysis results helped Ericsson process designers customize the candidate Agile practices, in such a way that leads to better compliance with their strategic objectives.

SAAF can be combined with most of the current frameworks of transition to Agile, and complement their lack of attention to the strategic aspects of the transition process. It can be also used as a stand-alone framework for strategic analysis of a set of candidate Agile practices, in order to find their potential compliance and conflicts with strategic interests of an organization. However, unlike most of the current frameworks of transitioning to Agile that use quantitative measure for managing the transition process, SAAF takes a *qualitative* goal-oriented approach for the analysis of Agile practices, as its purpose is not to measure the status of strategic objectives, hence it is basically aimed at specifying which set of objectives will be impacted by the introduction of new method, and how. Besides, anticipating the exact value to which a quantifiable strategic objective will be improved or degraded by an Agile practice needs strong evidential situations (i.e. evidences from very similar situations), which in most cases are not available.

8.1 Limitations and Threats to Validity of SAAF

The reliance of the framework on the evidential knowledge of Agile practices has both positive and negative aspects. On the positive side, it supports the strategic analysis with evidential situations from the experiences of other organizations in trying similar practices in their projects. Therefore, the analysis results are more reliable, compared to the cases that the basis of analysis is just the commonly known knowledge of Agile practices, or the experience of a process consultant. Besides, the strategic knowledge of Agile practices can help organizations in completing their Strategies Graph. The knowledge base proposes a *catalogue of strategic objective*, which can guide organizations in defining their own strategic objectives. However, relying on the evidential knowledge base has some negative aspects, such as lack of adequate information for some Agile practices, or the mismatch of organization situation with the information that is available in the knowledge base.

Threats to Validity. The framework was developed and has been applied in a single instance. In that instance, the organization had been under training for the transition to Agile for a couple of months before employing the SAAP framework. Therefore, the main threat to the internal validity of framework is that the middle managers, instead of unbiased expression of organizational strategic objectives, may have inadvertently focused on those objectives

which are somehow contributable by Agile practices. The following issues are also identified as threats to the external validity:

- *Incompleteness of Strategic Model* – the Strategies Graph that organizational managers develop might not represent all of the strategic objectives. To mitigate this risk, further reviews of Strategies Graph, in group and individual meetings, should be performed.
- *Lack of Similar Evidential situation* –due to the limitations of empirical studies and systematic review, not all of the organizational situations can be matched with exact evidence from the knowledge base. The severity of this issue is expected to be lessened as the content of knowledge base is expanded through further literature review.
- *Over-Pessimistic or Over-Optimistic Evaluations* – where there is no evidence for the contribution of an Agile practice to a strategic objective, yet the contribution is perceived possible, in some cases the subjective evaluations might be unrealistic. In fact, the consensus-based evaluation has limitations such as limited knowledge and experience of individual stakeholders, personal biases, representational adequacy of the membership of that consensus group, power and politics, etc.

The diversity of empirical studies, in terms of their research method (case study, ethnography, survey), and their approaches for validating their results are acknowledged as a risk to the inconsistency of repository data. Effects of this risk was tried to be minimized through the Data Analysis and Aggregation stage of the systematic review. Indeed, the repository provides the type of each empirical study, out of which a particular evidence is extracted.

SAAF proposes a new approach for extracting as-is process concerns, which is based on the use of *Itemized Strategic Dependency Models*. The proposed approach helps organizations to have a clear understanding of mutual dependency relations among their various organization roles, and the typical problems (as-is process concerns), which have been noted by organization actors. Although this approach showed quite successful in its real world trial at Ericsson, it is by no means claimed to be a complete as-is process assessment technique. Therefore, this method is not sufficient for organizations willing to go through heavyweight

process improvements initiatives, though it can be used as a complement to current process assessment techniques.

8.2 Future Work

Expanding the Process Experience Repository – So far the repository has been populated with the knowledge of 18 Agile practices, and 20 other Agile practices are under study. The website for this repository is being regularly updated to reflect the latest findings of empirical software engineering about Agile practices. The plan is to expand the repository to non-Agile practices. Work has started to collect and analyze Agile practices for requirements engineering (RE), Agile as well as non-Agile, and the partial results of work for two RE practices are published on the portal. The inclusion of non-Agile practices can be further improved by linking to the contents of current repositories of method fragments, such as *Open Process Framework* (OPF) (Firesmith) and the one of *Eclipse Process Framework* (EPF) (EPF, 2006).

Implementing Strategic Analysis Procedures – Analysis procedures of SAAF are all assumed to be done manually, while they have the potential be becoming automated. To this end, an attempt has been started to implement the *Strategic Contribution Analysis* and *Propagative Strategic Analysis*, as part of the services which are provided by ProcessExperience.org. An initial version of this system is now uploaded to the portal, which enables software organizations to build their Strategies Graph, and partly conduct the Strategic Contribution Analysis. The completion of this service can help the overall framework to run in shorter time.

Adding Strategic Trade-off Analysis to the Framework – besides the four strategic analysis procedures defined in Chapter 6, SAAF has the potential of incorporating the Strategic Trade-off Analysis. The strategic trade-off analysis of candidate practices is intended to guide the process of decision making among *alternative Agile practices*. In SAAF, alternative practices are compared with respect to their positive and negative contributions to the strategic goals of organizations. For instance, “Pair Programming” and “Peer Review” are two alternative practices that are often suggested for reducing defect rate in source code. However, depending to the project situation, there are other strategic goals which will be influenced differently by the enactment of any of these practices in an organization, e.g., “Cost of Development”, “Time

to Market”, “Productivity of Individuals”, and “Knowledge Sharing”. Strategic trade-off analysis of alternative practices can be carried out with respect to two classes of objectives:

1. *Built-in Objectives* – objectives which are studied to be contributable by an Agile practice. This information is available at the repository of Agile practices.
2. *Organizational Strategic Objectives* – objectives which have strategic significance for an organization.

The trade-off analysis of alternative CAPs becomes complex when we consider each practice within the context of its processes. Typically, Agile practices are proposed in sets of related fragments, which complement each other and assemble new candidate process. The trade-off analysis of two (or more) CAPs can be influenced by the strategic impacts of other Agile practices, which accompany each of alternative CAPs. The trade-off analysis of alternative processes can be carried out based on the result of aggregated strategic analysis.

Extending SAAF as a Generic Strategic Decision Support System – Strategic analyses procedures proposed by SAAF have the potential of being deployed in a wide range of applications, which require strategic decision making. As a general template, this can be applied on various situations that an organization would want to decide on a number of alternative solutions, or for any reason wants to evaluate the strategic impacts of a proposal. For instance, after the experience of SAAF for method evaluation in Ericsson, the R&D unit went through another initiative on *Innovation Management*, which was aimed at selecting a number of *Innovation Improvement techniques*, and introducing them to the organization. Since the organization had already developed its *Strategies Graph*, and were quite familiar with the principles of SAAF-based strategic analysis, they conducted a strategic evaluation of various *Innovation Improvement Techniques*, in order to choose the one which were best matching their strategic concerns. This evidence is supporting the claim that SAAF can be extended as a generic *Strategic Decision Support System*. However, at its current state, SAAF cannot be considered as a fully-fledged decision support system. In order to extend SAAF as a strategic decision making framework, there is need for wider literature review in Strategic Management and Business Administration.

Complementing SAAF with Post-Adoption Stages of Transitioning to Agile – Transitioning to a new development process is a gradual process, which does not happen over a

night. In particular in flexible process models, like Agile, what happens in most organizations is that the organization selects a small number of practices, enacts them in its development environment, and then based on the success or failure of those practice, would look for some new ones. Since the selection of new practices depends to the results of previous move, it is quite important to have a realistic perception of the strategic impacts of Agile practices that are already enacted within the development environment. SAAF has the capability of being complemented with *Post-Adoption Monitoring* services, which enables organizations to track the strategic impacts of their partial (or complete) transition to Agile. The primary pre-condition for the incorporation of such services to the SAAF, is to define *quantification measures* and *target values* of strategic objectives while defining the strategies graph (these attributes are now considered as option).

Integrating SAAF with other Agile Adoption Frameworks – The point of using SAAF is when an organization has already decided to make a transition to Agile, and has decided on a number of candidate Agile practices. However, it is possible that Agile not be a good solution for the process problems of the organization, and the decision of transitioning to Agile had me made incorrectly from the beginning. Moreover, once an organization made a right decision on transitioning to Agile, and selected a right set of practices, it still needs a set of guidelines for successfully implement Agile in its context. There are a number of Agile Adoption frameworks which can help software organizations in any of the above mentioned aspects. However, none of these frameworks are as comprehensive as SAAF in pre-adoption strategic analysis. One potential future work is to investigate the areas in which SAAF can be integrated with such frameworks, or ideas of these frameworks can be incorporated in the SAAF process to make a comprehensive Agile Adoption Framework.

Formalizing the Framework – One of the major factors in the design of SAAF was to be as lightweight as possible, in order not to impose an overhead to the process of transitioning to Agile. This design factor is mainly motivated by the intention of SAAF being used by practitioners. However, some aspects of SAAF can be further formalized, and be supported by specific ontologies. For instance, the concept of “situation” is informally explained for the repository elements, which makes it almost impossible to query Agile practices that satisfy a particular situation. Improving the situation specification by a particular ontology can facilitate

the tool support of the repository searches, and aligning the framework with the ideas of Situational Method Engineering.

Bibliography

- Ahonen, J., Forsell, M., & Taskinen, S.-K. (2002). A modest but practical software process modeling technique for software process improvement. In *Software Process: Improvement and Practice*, 7(1), 33-44.
- Amyot, D. (2003). Introduction to the User Requirements Notation: learning by example. In *Computer Networks*, 42(3), 285-301.
- Annie, I. A. (1996). Goal-Based Requirements Analysis. *Proceedings of the 2nd International Conference on Requirements Engineering (ICRE '96)*, IEEE Computer Society.
- Antón, A. I., & Potts, C. (1998). The use of goals to surface requirements for evolving systems. *Proceedings of the 20th international conference on Software engineering*. Kyoto, Japan, IEEE Computer Society.
- Aranda, J., Easterbrook, S., & Wilson, G. (2007). Requirements in the wild: How small companies do it. *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International*, 39-48.
- Aydin, M. N., Harmsen, F., van Slooten, K., & Stagwee, R. A. (2005). On the Adaptation of an Agile Information Systems Development Method. In *Journal of Database Management*, 16(4), 25-40.
- Bajec, M., Vavpotic, D., & Krisper, M. (2007). Practice-driven approach for creating project-specific software development methods. In *Information and Software Technology*, 49(4), 345-365.
- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001) Manifesto for Agile Software Development. accessed via: <http://agilemanifesto.org/>, in July 2009
- Birkhölzer, T., Chiniforooshan Esfahani, H., Dickmann, C., Vaupel, J., & Ast, S. (2011). Goal-Driven Evaluation of Process Fragments Using Weighted Dependency Graphs. *International Conference on Systems and Software Processes*, Springer, Berlin.
- Bloomfield, C. (2002). Bringing the Balanced Scorecard to Life: The Microsoft Balanced Scorecard Framework, Insightformation, Inc.
- Boehm, B., & Turner, R. (2003). *Balancing Agility and Discipline, a Guide for the Perplexed*, Addison-Wesley
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. In *Software, IEEE*, 22(5), 30-39.

- Bygstad, B., Ghinea, G., & Brevik, E. (2008). Software development methods and usability: Perspectives from a survey in the software industry in Norway. In *Interacting with Computers*, 20(3), 375-385.
- Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. In *European Journal of Information Systems*, 18, 332-343.
- Cares, C., Franch, X., Mayol, E., & Alvarez, E. (2006). Goal-Driven Agent-Oriented Software Processes. *32nd EUROMICRO Conference on Software Engineering and Advanced Applications*, 336-347, IEEE publications.
- Castro, J., Kolp, M., & Mylopoulos, J. (2001). A Requirements-Driven Development Methodology. K. D. Dittrich, A. Geppert and M. C. Norrie (Ed.). In: *Advanced Information Systems Engineering* (pp. 108-123), Springer
- Chiniforooshan, E. H., Yu, E., & Annosi, M. C. (2010). Itemized Strategic Dependency: a Variant of the i* SD Model to Facilitate Knowledge Elicitation. *4th International i* Workshop*.
- Chiniforooshan Esfahani, H., & Yu, E. (2010). A Repository of Agile Method Fragments. J. Münch, Y. Yang and W. Schäfer (Ed.). In: *New Modeling Concepts for Today's Software Processes (Proceedings of the ICSP'10)* (pp. 163 -174). Germany, Springer Berlin / Heidelberg
- Chou, S.-C., & Chen, J.-Y. J. (2000). Process program change control in a process environment. In *Software: Practice and Experience*, 30(3), 175-197.
- Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2000). *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers.
- Cohn, M., & Ford, D. (2003). Introducing an Agile Process to an Organization. In *Computer*, 36(6), 74-78.
- Colette, R. (1998). A Comprehensive View of Process Engineering. *Proceedings of the 10th International Conference on Advanced Information Systems Engineering*, Springer-Verlag, 1-24.
- Dardenne, A., Lamsweerde, A. v., & Fickas, S. (1993). Goal-directed requirements acquisition. In *Science of Computer Programming*, 20(1-2), 3-50.
- Darimont, R., Delor, E., Massonet, P., & Lamsweerde, A. v. (1997). GRAIL/KAOS: an environment for goal-driven requirements engineering. *Proceedings of the 19th international conference on Software engineering*. Boston, Massachusetts, United States, ACM, 453-462.
- Derniame, J. C., Kaba, B. A., & Wastell, D. G., Eds. (1999). *Software Process: Principles, Methodology, Technology*. LNCS 1500, Springer-Verlag.
- Eilat, H., Golany, B., & Shtub, A. (2008). R&D project evaluation: An integrated DEA and balanced scorecard approach. In *Omega*, 36(5), 895-912.
- Elahi, G., Yu, E. S. K., & Annosi, M. C. (2008). Modeling Knowledge Transfer in a Software Maintenance Organization - An Experience Report and Critical Analysis. *the proceeding of the IFIP WG 8.1 Working Conference on Practice of Enterprise Modeling (PoEM'08)*, 12-13.

- EPF. (2006). "Eclipse Process Framework Project (EPF)." Retrieved jun, 2009, from <http://www.eclipse.org/epf/>.
- Firesmith, D. Open Process Framework (OPF), <http://www.opfro.org/>
- Florac, W. A., E., P. R., & Carleton, A. D. (1997). Practical Software Measurement: Measuring for Process Management and Improvement. CMU/SEI-97-HB-003, Software Engineering Institute
- Gill, A. Q. (2011). *Enterprise Agile Adoption and Improvement: A Comprehensive Information Guide to Agile System Development Strategy and Implementation*, LAP Lambert Academic Publishing
- Gonzales-Baixaui, B., Prado Leite, J. C. S., & Mylopoulos, J. (2004). Visual variability analysis for goal models. *Proceedings. 12th IEEE International Requirements Engineering Conference*, 198-207.
- Gonzalez-Perez, C., Giorgini, P., & Henderson-Sellers, B. (2007). Method Construction by Goal Analysis. *Proceedings of Int. Conf. on Information System Development*, 79-91, Springer US.
- Gorschek, T., & Wohlin, C. (2004). Packaging software process improvement issues: a method and a case study. In *Software: Practice and Experience*, 34(14), 1311-1344.
- Grau, G., Franch, X., & Maiden, N. A. M. (2008). PRiM: An i*-based process reengineering method for information systems specification. In *Information and Software Technology*, 50(1-2), 76-100.
- Grau, G., Franch, X., Mayol, E., Ayala, C., Cares, C., Haya, M., Navarrete, F., Botella, P., & Quer, C. (2005). RiSD: A Methodology for Building i* Strategic Dependency Models. *Proceedings of The Seventeenth International Conference on Software Engineering and Knowledge Engineering (SEKE'05)*, 259-266.
- Grembergen, V. W., & Amelinckx, I. (2002). Measuring and managing e-business projects through the balanced scorecard. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* 9 pp.
- Harmsen, A. F., & Saeki, M. (1996). Comparison of four method engineering languages. S. Brinkkemper, K. Lyytinen and R. J. Welke (Ed.). In: *Principles of method construction and tool support* (pp. 209-231). London, Chapman & Hall
- Henderson-Sellers, B. (2003). Method engineering for OO systems development. In *Communications of the ACM*, 46(10), 73-78.
- Horkoff, J., & Yu, E. (2009). A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models. *Proceedings of CEUR Workshop in CAiSE,09*.
- Huang, C. D., & Hu, Q. (2007). Achieving IT-Business Strategic Alignment via Enterprise-Wide Implementation of Balanced Scorecards. In *Information Systems Management*, 24(2), 173 - 184.
- Humphrey, W. S. (1989). *Managing the software process*, Addison-Wesley Longman Publishing Co. Inc.
- Jansen, S., Brinkkemper, S., & Finkelstein, A. (2007). Providing Transparency In The Business Of Software: A Modeling Technique For Software Supply Networks. L. Camarinha-Matos, H.

- Afsarmanesh, P. Novais and C. Analide (Ed.).In: *Establishing The Foundation Of Collaborative Networks* (pp. 677-686). Boston, Springer
- Janzen, D. (2009). SEED: Software Engineering Evidence Database.
- Jones, M., & Karsten, H. (2008). Giddens's structuration theory and information systems research. In *MIS Quarterly*, 32(1), 127-157.
- Kaiya, H., Horai, H., & Saeki, M. (2002). AGORA: attributed goal-oriented requirements analysis method. *Proceedings of IEEE Joint International Conference on Requirements Engineering*, 13-22, IEEE Computer Society
- Kaplan, R. S., & Norton, D. P. (1996a). *The balanced scorecard: Translating strategy into action*. Boston, Harvard Business School Press.
- Kaplan, R. S., & Norton, D. P. (1996b). Using the Balanced Scorecard as a Strategic Management System. In *Harvard Business Review*, 74, 75-85.
- Kaplan, R. S., & Norton, D. P. (2004). *Strategy Maps : Converting Intangible Assets into Tangible Outcomes*. Boston, Harvard Business School.
- Kavakli, V., & Loucopoulos, P. (1999). Goal-driven business process analysis application in electricity deregulation. In *Information Systems*, 24(3), 187-207.
- Ketter, W., Banjanin, M., Guikers, R., & Kayser, A. (2009). Introducing an Agile Method for Enterprise Mash-Up Component Development. *Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*, IEEE Computer Society.
- Kim, J., Suh, E., & Hwang, H. (2003). A model for evaluating the effectiveness of CRM using the balanced scorecard. In *Journal of Interactive Marketing*, 17(2), 5-19.
- Kitchenham, B. (2004). Procedures for Performing Systematic Reviews.TR/SE-0401,Software Engineering Group, Department of Computer Science, Keele University
- Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004). Evidence-Based Software Engineering. *Proceedings of the 26th International Conference on Software Engineering*, IEEE Computer Society.
- Krajewski, L., Ritzman, L., & Malhotra, M. (2009). *Operations Management (9th Edition)* Prentice Hall.
- Krasteva, I., Ilieva, S., & Dimov, A. (2010). Experience-Based Approach for Adoption of Agile Practices in Software Development Projects. (Ed.).In: *Advanced Information Systems Engineering* (pp. 266-280), Springer Berlin / Heidelberg
- Lamsweerde, A. v. (2001). Goal-oriented requirements engineering: a guided tour. *Proceedings of fifth IEEE International Symposium on Requirements Engineering*. , 249-262.
- Lamsweerde, A. v. (2003). From System Goals to Software Architecture. (Ed.).In: *Formal Methods for Software Architectures* (pp. 25-43)
- Lamsweerde, A. v. (2004). Goal-oriented requirements engineering: a roundtrip from research to practice. *Proceedings of 12th IEEE International Requirements Engineering Conference*, 4-7.

- Lamsweerde, A. v., & Letier, E. (2004). From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering. (Ed.).In: *Radical Innovations of Software and Systems Engineering in the Future* (pp. 153-166), Springer, Berlin-Heidelberg
- Lapouchnian, A., Yu, Y., & Mylopoulos, J. (2007). Requirements-Driven Design and Configuration Management of Business Processes. *5th International Conference on Business Process Management*, 246-261, Springer, Berlin.
- Letier, E., & Lamsweerde, A. v. (2002). Agent-based tactics for goal-oriented requirements elaboration. *Proceedings of the 24th International Conference on Software Engineering*. Orlando, Florida, ACM.
- Letier, E., & Lamsweerde, A. v. (2004). Reasoning about partial goal satisfaction for requirements and design engineering. *Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering*. Newport Beach, CA, USA, ACM, 53 - 62.
- Martin, R. H., & Raffo, D. (2000). A model of the software development process using both continuous and discrete models. In *Software Process: Improvement and Practice*, 5(2-3), 147-157.
- Martinsons, M., Davison, R., & Tse, D. (1999). The balanced scorecard: a foundation for the strategic management of information systems. In *Decision Support Systems*, 25(1), 71-88.
- McFeeley, B. (1996). IDEAL: A User's Guide for Software Process Improvement. CMU/SEI-96-HB-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA
- Mikulenas, G., & Kapocius, K. (2011). An Approach for Prioritizing Agile Practices for Adaptation. (Ed.).In: *Information Systems Development* (pp. 485-498), Springer New York
- Mujeeb-u-Rehman, M., Xiaohu, Y., Jinxiang, D., & Abdul Ghafoor, M. (2005). Heterogeneous and homogenous pairs in pair programming: an empirical analysis. *Canadian Conference on Electrical and Computer Engineering*, 1116-1119, IEEE Press.
- Niazi, M. (2006). Software Process Improvement: A Road to Success. (Ed.).In: *Product-Focused Software Process Improvement* (pp. 395-401). Berlin Springer
- Nilsson, N. (1971). *Problem Solving Methods in Artificial Intelligence*, McGraw Hill.
- O'Donnell, M. J., & Richardson, I. (2008). Problems Encountered When Implementing Agile Methods in a Very Small Company. (Ed.). In: *Software Process Improvement* (pp. 13-24). Berlin Heidelberg, Springer
- Oshiro, K., Watahiki, K., & Saeki, M. (2003). Goal-oriented idea generation method for requirements elicitation. *Proceedings, 11th IEEE International Requirements Engineering Conference*, 363-364.
- Osterweil, L. (2005). Unifying Microprocess and Macroprocess Research. (Ed.).In: *SPW 2005* (pp. 68-74). Heidelberg, Springer
- Persson, A., & Stirna, J. (2001). EKD User Guide, deliverable D3. *IST Program Project IST-2000-28401HyperKnowledge*, KTH, Sweden.

- Pettersson, F., Ivarsson, M., Gorschek, T., & Öhman, P. (2008). A practitioner's guide to light weight software process assessment and improvement planning. In *Journal of Systems and Software*, 81(6), 972-995.
- Porter, M. (1996). What is Strategy? In *Harvard Business Review*, 74(6), 61-78.
- Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. In *Journal of Systems and Software*, 81(11), 1899-1919.
- Ralyté, J., & Rolland, C. (2001). An Assembly Process Model for Method Engineering. (Ed.).In: Proceedings of CAiSE '01 Proceedings of the 13th International Conference on Advanced Information Systems Engineering, *Advanced Information Systems Engineering*, Springer-Verlag London, UK, 267-283
- Rohunen, A., Rodriguez, P., Kuvaja, P., Krzanik, L., & Markkula, J. (2010). Approaches to Agile Adoption in Large Settings: A Comparison of the Results from a Literature Analysis and an Industrial Inventory. (Ed.).In: *Product-Focused Software Process Improvement*, Springer Berlin / Heidelberg, 77-91
- Rossi, M., Ramesh, B., Lyytinen, K., & Tolvanen, J. (2004). Managing Evolutionary Method Engineering by Method Rationale. *Journal of the Association for Information Systems*, 5(9), 356-391.
- Saeki, M. (2003). CAME: The first step to automated method engineering. *Workshop on Process Engineering for Object-Oriented and Component-Based Development*, 7--18.
- Savolainen, P., Sihvonen, H.-M., & Ahonen, J. (2007). SPI with Lightweight Software Process Modeling in a Small Software Company. P. Abrahamsson, N. Baddoo, T. Margaria and R. Messnarz (Ed.).In: *Software Process Improvement* (pp. 71-81). Berlin, Springer
- Schwaber, K. (1995). Scrum Development Process. *OOPSLA'95 Workshop on Business Object Design and Implementation*.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Redmond, WA, USA, Microsoft Press.
- SEI (2006). CMMI® for Development, Version 1.2. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- SEI (2009). CMMI Process Maturity Profile Software Engineering Institution, Carnegie-Mellon University, Pittsburgh, PA
- Seunghun, P., Hyeonjeong, K., & Doo-Hwan, B. (2009). Change Impact Analysis of a Software Process Using Process Slicing. *Quality Software, 2009. QSIC '09. 9th International Conference on*, 359-366.
- Sidky, A., & Arthur, J. (2007). Determining the Applicability of Agile Practices to Mission and Life-Critical Systems. *Proceedings of the 31st IEEE Software Engineering Workshop*, 3-12, IEEE Computer Society.
- Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: the agile adoption framework. In *Innovations in Systems and Software Engineering*, 3(3), 203-216.

- Slaughter, S. A., Levine, L., Ramesh, B., Pries-Heje, J., & Baskerville, R. (2006). Aligning Software Processes with Strategy. In *MIS Quarterly*, 30, 891-918.
- SMG (2010). The Strategic Management Maturity Model, Strategy Management Group, Balanced Scorecard Institute, North Carolina
- Soffer, P. (2005). Scope analysis: identifying the impact of changes in business process models. In *Software Process: Improvement and Practice*, 10(4), 393-402.
- Stirna, J., & Persson, A. (2007). Ten Years Plus with EKD: Reflections from Using an Enterprise Modeling Method in Practice. *Eleventh International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD'07), held in conjunction with CAiSE 2007*. Trondheim, Norway.
- Stirna, J., Persson, A., & Sandkuhl, K. (2007). Participative Enterprise Modeling: Experiences and Recommendations. (Ed.). In: *Advanced Information Systems Engineering* (pp. 546-560), Springer Berlin
- Sureshchandra, K., & Shrinivasavadhani, J. (2008). Adopting Agile in Distributed Development. *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*, 217-221.
- Susi, A., Perini, A., & Mylopoulos, J. (2005). The Tropos Metamodel and its Use. In *Informatica Journal*, 29, 401--408.
- Szalvay, V., Mar, K., & James, M. (2008). Agile Transformation Strategy, Danube Technologies, Inc.
- Tague, N. R. (2004). "Seven Basic Quality Tools." *The Quality Toolbox*. Retrieved November 2010, from <http://asq.org/learn-about-quality/cause-analysis-tools/overview/fishbone.html>.
- Tran, Q., & Chung, L. (1999). NFR-Assistant: tool support for achieving quality. *Application-Specific Systems and Software Engineering and Technology, 1999. ASSET '99. Proceedings. 1999 IEEE Symposium on*, 284-289.
- Yu, E., & Mylopoulos, J. (1994). Understanding "why" in software process modelling, analysis, and design. *Proceedings of the 16th international conference on Software engineering*. Sorrento, Italy, IEEE Computer Society Press.
- Yu, E. S. K. (1997). Towards modelling and reasoning support for early-phase requirements engineering. *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, 226-235, IEEE Computer Society.
- Yue, K. (1987). What Does It Mean to Say that a Specification is Complete? *Proceeding of IWSSD-4, Fourth International Workshop on Software Specification and Design*.
- Z.151 (2008). *Goal Requirements Language (GRL)*. Geneva, ITU-T Recommendation.

Appendix 1. SAAF Evaluation Questionnaire

Name:

Organization Role:






Familiarity with the initial analysis:

The purpose of this evaluation process is to investigate the perceived correctness of the analysis results of the SAAF. The analysis results had anticipated the potential impacts of a set of agile practices on The strategic objectives of the organization. Now that more than a year is passed over the adoption of agile practices, we would like to investigate the degree of the correctness of our pre-adoption analysis results.

Your participation in this process is sincerely appreciated.

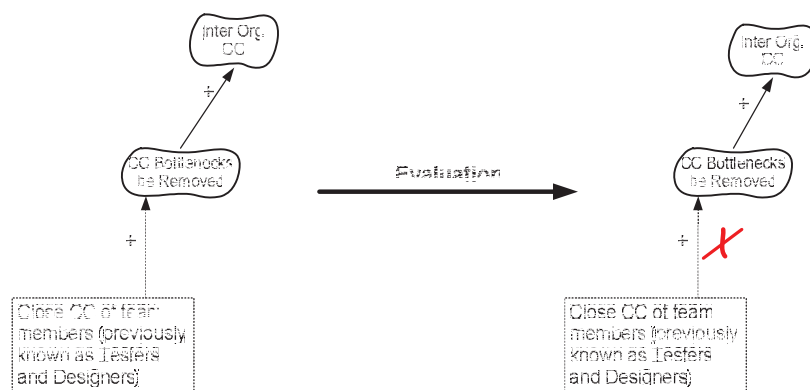
Instructions:

In this evaluation form, you are requested to express your opinion about the correctness of above items. The evaluation process is based on a five scale qualitative measure, based on which you express your degree of agreement with the analysis results:

| | | |
|---|--------------------------|-------------------|
|  | <input type="checkbox"/> | Strongly Agree |
|  | <input type="checkbox"/> | Partly Agree |
|  | <input type="checkbox"/> | No Opinion |
|  | <input type="checkbox"/> | Partly Disagree |
|  | <input type="checkbox"/> | Strongly Disagree |

For every agile practice, this document first provides a short description, and then presents its impact over organization concerns and its contributions to the organization objectives. Please mark any analysis item with your degree of agreement.

For instance, for the for the Agile practice “Scrum Team Structure” we have analyzed that it will make a positive contribution to the removal of Communication Collaboration (CC) bolltlenecks (Figure 1 - Left). In evaluation process, participants express their opinion about this result; for instance they my mark the contribution with “Disagree” mark; meaning that they do not believe that such agile fragment has caused such effect in the organization (Figure 1 – Right):



Please note that only contribution relations, which are coming from the agile practices to the Strategies Graph are being evaluated. These relations are represented as dotted links.

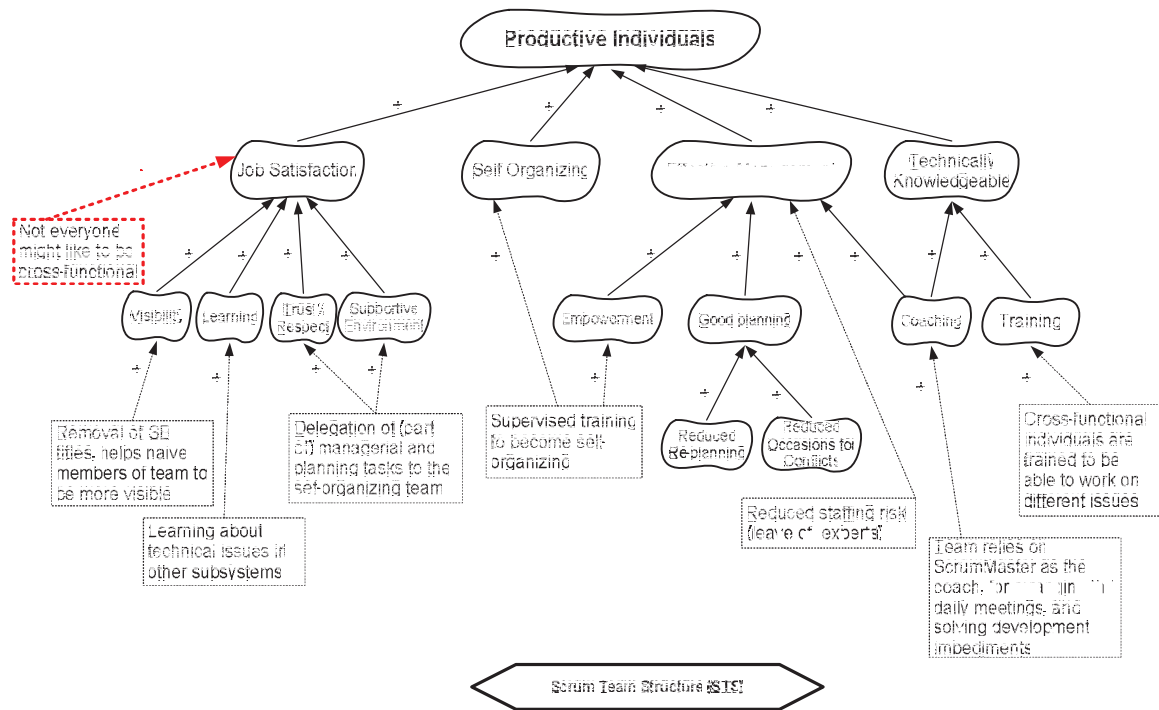
- It is not needed to evaluate all of the links
- Please put the evaluation mark next to each dotted link.

Scrum Team

Unlike other agile methodologies that provide a set of technical AMFs, Scrum is mainly composed of a set of managerial AMFs, which are about the arranging people, handling requirements, and planning releases. Therefore the integral part of Scrum methodology is the team structure that it proposes for software organization. We would like to highlight the fact that the failure of an organization in adopting Scrum team structure, is somehow the same as failure in adopting the whole methodology. In an Scrum environment there are three primary roles: ScrumMaster, Product Owner, and the Team. Here, we focus on the Scrum Team, which is typically known as:

- A group of individuals, working together to achieve sprint goals
- Scrum team is usually 6~10 people
- Scrum team do not include any of the conventional software engineering roles, such as programmer, designer, tester, architect, . . .
 - This does not mean that people must perform tasks for which they have no expertise. In fact, in most cases individuals will spend most of their time working on disciplines they worked before adopting Scrum.
 - But, individuals are expected to work beyond their preferred disciplines, whenever doing so would be of interest of the team.
- Scrum team members are expected to be
 - Self-organizing
 - Cross-functional

To what extent you agree with the impacts of this agile practice on strategic goals of organization, as shown here?



Similar questions were asked for this candidate agile practice with respect the other portions of the organization's strategies graph. However, due to confidentiality reasons, the rest of evaluation, helps naive models cannot be displayed here. Similar, approach was also used for evaluating the analysis results of other candidate Agile practices.

