Using Rational Unified Process in an SME - A Case Study

Conference Paper in Lecture Notes in Computer Science · November 2005 DOI: 10.1007/11586012_14 · Source: DBLP CITATIONS READS 6,082 11 3 authors: Geir Kjetil Hanssen Hans Westerheim SINTEF SINTEF 98 PUBLICATIONS 1,910 CITATIONS 22 PUBLICATIONS 150 CITATIONS SEE PROFILE SEE PROFILE Finn Olav Bjørnson SINTEF 38 PUBLICATIONS 971 CITATIONS SEE PROFILE

Using Rational Unified Process in an SME – A Case Study

Geir Kjetil Hanssen¹, Hans Westerheim¹, Finn Olav Bjørnson²

¹SINTEF ICT, N-7465 Trondheim, Norway {geir.kjetil.hanssen, hans.westerheim}@sintef.no

²Norwegian University of Science and Technology, N-7491 Trondheim, Norway bjornson@idi.ntnu.no

Abstract. The Rational Unified Process (RUP) is a comprehensive software development process framework emphasizing use-cases, architecture focus and an iterative approach. RUP is widely known and many organizations have tried to adopt it. Being a framework, RUP has to, in some way, be tailored to the specific context of use, no software development project is alike. This paper presents a case study of a Norwegian SME that tried to adopt RUP in the simplest way, by introducing the methodology by providing comprehensive documentation and some simple training. Our study shows that the use of RUP had some positive effects but also that the use has been scattered. Interviews with users of RUP show that there is a great need of better training and practical support in getting most value out of RUP. The key message is that if you consider taking RUP into use you have to invest resources in it. Training and support are key success factors.

1. Introduction

The Rational Unified Process (RUP) is a software development process framework consisting of a more or less complete set of process elements for software development projects [1]. RUP defines a software development project as a set of disciplines, e.g. requirements handling, implementation etc., running from start to end through the whole project life cycle divided in a set of project phases. A project is performed by a group of actors, each having one or more well defined roles. Each role participates in one or more activities producing one or more artifacts. A discipline can run in iterations, that is, repetitions within a phase. Activities, roles and artifacts are the basic process elements of RUP. RUP is a prescriptive and plan driven methodology. As RUP is a comprehensive framework covering most aspects of a software development process it means that it in some way must be adapted to the situation of use, either ad-hoc for each project or in advance to produce a company wide standard.

In this paper we present a case study that describes the use of RUP in a company where no restrictions or guidelines were put on the use of RUP. The project managers and senior developers were given courses in RUP, and RUP Online (an electronic process guide on web) was purchased and installed. No common guidance for the use of RUP in projects was given. The company had no defined goals for introducing RUP; it was basically based on a belief that RUP would increase the professionalism

in the company. The study has been conducted within a smaller Norwegian software development company. Three researchers have followed the company during a period of three years. This paper describes the experience from using RUP and derives some key conclusions that may be of use for others considering the use of RUP.

The paper has the following structure:

- The research method is described (collection of empirical data and data analysis).
- The research context of the case study, that is, the company, is described.
- The results part documents information and data collected. This includes
 descriptions of the usage of RUP as well as elements laying the ground
 for the forthcoming tailoring of RUP. Further on the results from the
 analysis of four projects and five interviews are documented.
- A discussion trying to clarify the key points from the analysis and giving a conclusion.

2. Research Method

2.1 Data collection

The study has taken the form of a case study [2]. The research has been conducted by three external researchers mainly using project managers and software developers from the company as a source for data.

The first set of data was collected by interviewing project managers representing four projects. Prior to this series of interviews, the researchers prepared a spreadsheet that had a row for each role, activity and artifact described by RUP, grouped by the disciplines defined by RUP. A column was allocated to each project. The researcher conducting the interview asked the project manager about the use of every single role, activity and artifact in the actual project. If the element was used in the project as described by RUP, the actual cell was colored. If the item was used as described by RUP, but changed or replaced by a tailored element, the cell was colored and a comment was written about the change from the original item description in RUP. If the element was not used at all in the project, the cell was left blank.

The second set of data was collected by the means of semi-structured interviews with five other employees (each having experience with RUP from several various projects). The respondents had the following main responsibilities: 1) developer, 2) developer/project manager, 3) developer/project manager/test manager, 4) project manager/requirements engineer and 5) customer contact. Prior to the interviews, the researchers developed an interview guide. The guide consisted of questions with focus on their personal experience with using RUP across multiple projects to

document a broader experience. The guide was open, allowing the respondents to freely discuss their experience.

These interviews were recorded and transcribed by the researchers. The transcriptions were reviewed by the interviewed objects, and possible corrections and clarifications were made.

2.2 Data Analysis

The spreadsheet documenting the use of RUP and the transcribed interviews were basis for the data analysis. From the beginning it was clear that the researchers were to use qualitative data analysis methods due to the nature of the data collected [3, 4].

Analysis of the spreadsheet. The spreadsheet was printed in 25% size of its actual size. This was done to get an *overview* of the RUP usage. The overview gave a clear visual picture of what parts of RUP which were really used. The comments in the spreadsheet were read through. The researchers tried to match comments with the non-use, to see if there might be some statements supporting the lack of usage. The RUP usage for the projects were also compared to the project definition, the scope of the project and the type of customer as a starting point for an understanding of the actual use and non-use of RUP elements.

Analysis of the interviews. The researchers used the constant comparison method [3] to identify the factors affecting the use of RUP among the project managers and senior developers. All the transcriptions were printed out, and each of the researchers got one copy each. The single transcriptions were read individually, and the researcher tagged statements in the documents which said something about use or non-use of RUP, reasons for using or not using RUP, and also positive and negative aspects with RUP itself. Then the researchers had a common work shop where the individual tagging were put onto a white board, and then compared. The comparison was the basis for a common summary of the interviews.

The main motivation for selecting this approach to data collection and data analysis was that the researchers did not have any pre-information about the use of RUP, and therefore no assumptions or hypothesis to test, thus a qualitative approach seemed appropriate. This motivation was supported by the relative low number of data points available from such a small company.

3. Research context

The company described in this case is today a Norwegian software consultancy company with 50 employees, located in two different geographic offices.

They are mainly developing software systems with heavy back-end logic and often with a web front-end, typically portals. However, they also develop lighter solutions with most emphasis on the front-end.

4 Geir Kjetil Hanssen, Hans Westerheim and Finn Olav Bjørnson

The company acts as an independent software supplier, though there are close relationships to the biggest customers. Of the 50 employees today, 35 are working as software developers. Java and J2EE are used as the main development platform. The domain of which the company develops software is mainly for the banking and finance sector, as well as for public sector. The company has run 50 development projects within the bank and finance sector the last twelve years, and about 30-40 projects within the public sector the last 15 years.

Four employees are certified RUP-mentors acting as advisors in other SW-organizations, in addition to this they also used to run training courses in RUP as part of their partnership with Rational (now IBM Rational).

During the work described in this paper the company was declared bankrupt, and then restarted with new owners, but with the same employees. The data collection (using the spreadsheet) took place before the bankruptcy; the interviews took place about six months after the company was restarted.

4. Results

4.1 Interview round 1: Documenting the use of RUP

The four projects investigated had a scattered use of RUP. Interviewing the project leaders we documented the projects per phase to see which process elements were used and which were not and the corresponding reasons for that. In the following we present a summary per phase for each project (named project A to D).

The business modeling discipline

Project A was about porting functionality, no new functionality was introduced, thus not needing business modeling. For project B the customer had provided a business use case that was sufficient. Project C was developing software to be integrated with other systems. The business modeling discipline was used to clarify these interfaces. Project D had a business modeling discipline although it was not performed exactly as described by RUP.

The requirements discipline

Project A used the discipline partly to specify requirements for how to join the user interface of several systems. The other three projects used the requirements discipline quite extensively.

The analysis and design discipline

The elements in the discipline were partly used for all four projects. However there was a lot of adoption to the project context.

The implementation discipline

The use of the process elements in the implementation discipline was scattered. Although all four projects used it, project A used it briefly, project C and D used it extensively.

The test discipline

Project B and D had an extensive use of the process elements in the test discipline while the two other projects did not follow RUP for testing.

The deployment discipline

Project A had deployment activities but these were not done according to RUP. Project B had at the time of the interview not got to this. The deployment in project C was done partly by the customer that had responsibility for most of the activities. Project D did utilize most elements from RUP.

The configuration and change management discipline

Project A did not follow RUP at all; this was however done using a specialized system guiding the process of configuration and change management. Project B and C did use RUP pretty extensively for this discipline. For project D the customer handled this responsibility following other procedures than described by RUP.

The project management discipline

Project A did not follow RUP except for the use of the software architect role. Project B and C did use most of the process elements from RUP. In the case of project D the customer had the project management responsibility themselves.

The environment discipline

Project A had merely no use of this discipline. Project B used most process elements. Project C used only a few but project D used several.

By mapping the use of process elements for the four projects we made a visual map documenting the use of each process element in RUP, ordered by the eight disciplines that RUP describes (see fig. 1).

Inclinence Process Analysis Concept Study Royal Process Concept Study Royal Royal Process Concept Study Royal Process Concept Study Royal Royal Royal Process Concept Study Royal Royal Royal	es		_		Ro	oles					Roles				_
Integration Description De		ш				Implementer								Ш	L
Scheller Scheller Content Edeate Co		Ш				Software Architect					Change Control I	Manager		Ш	L
Code Reviewer Teature Specification	usiness Model Reviewer	П				Integrator					Project Manager				Γ
primers large Dearwes Specification Tester Americas Americas	acts					Content Editor	Т				Integrator				Γ
Jasen Production Manager Jasen Production M		П	Т	\neg		Code Reviewer					Software Architec	t			Г
Test Care Model arises Criticis arises Critici		H	-	+				Н						Н	۲
Arrefacts		Н	_	\blacksquare		_	+							ш	_
Implementations Subsystem Science Reagonationy Interest Workers Reagonationy Interest Workers Reagonationy Interest Workers Interest Subsystem Stephen Interest Workers Interest Subsystem Subsystem Interest Workers Interest Subsystem Subsystem Interest Subsystem Subsystem Interest Subsystem Inte		ш	_	ш										_	
suriesses Morkers grantsetors Workers lites Software Anchibecture Concurrent grantsetors Workers lites Component Common Vocabulary Creat Component Common Vocabulary Common V	-	ш			Ar					_					Ĺ
agrossorand Unit links Information Model Implementation Model I	siness Entities	1				Implementation Subsystem					Project Resposito	ory		1	ı
processor Unit New Common Vocabulary planes a Common Vocabulary planes and the Common Vocabulary planes and	usiness Use Case Realization					Software Architecture Document					Workspace (Inter	gration)			Γ
grantestonal Unit glaces a Common Vocabulary glaces a Common Vocabulary glaces a Common Vocabulary Grant Schore Activities Record Schore Activities As a Component Activities Structure the Substrees List Cases Model Recurrent Component Feel Component Feel Component Feel Component From Project Component and Change Final Project Component an	isiness Workers	П	_	П		Implementation Model	\top				Work Order		T	П	Ē
illes Interes	rganisational Unit	\vdash	\dashv	-		Integration Build Plan	-	Н		_	Workspace (Dev	elonment)	+		Г
rest Coresponent Floridation of Business Declared Business Declared State Case Model Floridation State Case Model Floridation State Case Medium State	•	ш		ш		_	_					olopinont)	+	Н	۲
Activities Review Record Build Build Review Record Build Build Review Record Build Build Activities Review Record Build Build Build Activities Review Record Build B		_		_		_ '							\perp	ш	L
Lincture the Business Use Case Model and Business Workers and Case Speciation and Case Speciation and Case Speciation and Case Speciation and Case Model and Case Model and Case Speciation and Case Model and Case Model and Case Speciation and Case Speciation and Case Model and Case Model and Case Model and Case Speciation and Case Model and Case Model and Case Speciation and Case Speciation and Case Model and Case Model and Case Speciation and C		ш	_	ш			\perp							ш	L
setal a Business Worker statal a Business Worker statal a Business Worker statal a Business Extend provided the Superior State of Subsystem provided the Subsystem provided the Superior State of Subsystem provided the Subsystem provided		ш									Project Measurer	nents		Ш	L
and Business Worker that a Business Worker that Busine	ructure the Business Use Case Model	1 1				Build					Change Request	S			ı
Table 1 Business Worker Table 2 Business Libra Clase Model Throw the Business Libra Clase Model Throw throw the Business Libra Clase Model Throw throw the Business Libra Clase Model Throw thr	etail a Business Use Case				Ac	tivities					Activities				
Table 1 Business Worker Table 2 Business Libra Clase Model Throw the Business Libra Clase Model Throw throw the Business Libra Clase Model Throw throw the Business Libra Clase Model Throw thr	nd Business Workers	П		П		Structure the Implementation Model					Plan Project Con	figuration and Change			Г
Intellegence Components Intellegence Components Integrate and Subsystem Integrate and In		H	_	Н		-	+						+	Н	г
winew the Business Use Case Model winemase Signed Model stem Analyst stem A B C D stem Analyst stem A B C D Test Test Case Specifier ser Interface Designer horized test Secretic Designer test Secr		\vdash	+	+		-		_					+	Н	۲
winderments A B C D Stem Analyst stem Ana	·	\vdash	_	ш									\perp	ш	H
Interest As B c D Roles Interest Analyst Interest As B c D Roles Int		ш	┸	ш		-	\perp	Ш					\perp		Ĺ
setern Analyst ac Case Specifier for Interface Designer chicket costay Specifier for Integration integr	eview the Business Object Model	آليا	┸	Ш		Integrate the system	الل						آلل		L
Roles Ro	uirements	A	ВС	D	Te	est	Α	В	C	D	Manage Change	Requests	П		ſ
stem Analyst e class Specifier State-holder Sta					Ro	oles					Project Manage	ment	A	В	c
the Case Specifier interface Designer integrated integ										\neg					1
Test Designer Integrator Implementer I			-				+		H	\dashv		ist	т 1	$\overline{}$	г
Integrator Impelmenter Sossary		ш	_				+		Н	_		iot	Н	Н	۲
Implementer Arrafects Feet Plan Feet Automation Architecture Frest Request Suprements Management Plan Suprements Attributes Test Surity Change Request Test Case Ingration Plan Software Development Plan Software Architect Reviewer Test Surity Specification Test Surity Test Surity Test Surity Test Surity Change Request Test Surity		ш	_			-	+		Ш				\vdash	ш	L
Artefacts Feet Plan Test Automation Architecture Test Suddelines Sukenbolder Request Sukenbolder Request Jayze the Problem Jayze the Probl						-	\perp		Ш	_			Ш		
Test Automation Architecture Guernements Management Plan quirements Attributes Test Guidelines Test Guidelines Test Guidelines Test Environment Configuration Test Script Test Evaluation Summary Test Evaluation Summary Test Evaluation Summary Test Evaluation Summary Test Suste Test Buta Test Suste Test Data Test Suste Test Data Test Suste Change Request Test Suste Test Suste Test Suste Test Suste Change Request Test Suste Test Suste Test Suste Test Suste Test Suste Change Request Test Suste Tes	acts					Implementer	L.		الـــا		Software Architec	et			Ĺ
recase Model quiements Authorites quiements quiement	ossary				Ar	tefacts					Process Enginee	r	П	П	ſ
e-Case Model quirements Management Plan quirements Authoriuses sixeholder Request quirements Authoriuses quireme	sion	\sqcap				Test Plan					Test Designer		П		Ī
regularements Management Plan quidernents Management Plan quidernents Alterbuses		\vdash							Н				Н		r
reast Environment Altributes ketholder Request poplementary Specifications ties Test Stute Test Suste Test		H	-				+						-	_	_
Test Script Test Script Test Script Test Script Test Script Test Data Test Data Test Data Test Suite Change Request Test Suite Test Suite Test Suite Test Suite Test Suite Test Data Test Suite Change Request Test Case Workload Model Issues List Test Gase Workload Model Issues List Test and Suite Suite Test Suite Change Request Workload Model Issues List Test Case Workload Model Issues List Test and Evaluation Nesson Verify Test Approach Verify Test App	-	\vdash	_	ш			+			_				_	_
potementary Specifications Ities Test Evaluation Summany Test State Test Sta	•	ш		Ш			\perp							ш	L
Activities alogical process of the System of	akeholder Request	ш				Test Script					Risk List				L
derstand Stakeholder Needs fire the System fire the Architecture froduct Artwork beployment Unit bevelop Support Material fire the Architecture froduct Artwork beployment Unit bevelop Support Material fire the System	pplementary Specifications					Test Evaluation Summary					Business Case				Ī
alyze the Problem derstand Stakeholder Needs ince the System anage the Scope of the System anage Request anage Request Test Case Danage Request Test Case Problem Resourments Prior Result Acceptance Plan Project Measurements Prior Result Acceptance Plan Project Measurements Prior Result Acceptance Plan Project Measurements Prior Result Acceptance Plan Problem Resolution Plan Risk Management Plan Change Request Problem Resolution Plan Risk Management Plan Risk Managemen	ities					Test Results	\top				Iteration Plan		T		Г
Idenstand Slakeholder Needs Ifrine the System Irraige the Scope of the System Irraige the Scope of the System Irraine the System Definition Irraine Specification Irvaire Architect Reviewer Irvaire Architect Reviewer Irvaire Architect Irraine Irvaire Architect Irraine Irraine Specification Irvaire Architect Irraine Ir						Test Data	+				Software Develor	ment Plan	+	П	Г
change Request Test-ideas List Workload Model Issues List Issues L	-	$\boldsymbol{\vdash}$	_	-			+	_					+	Н	۲
Test-ideas List Test-deas List Novikload Model Lisus List List List List List Define Evaluation Mission Varidate Build Stability List		$\boldsymbol{\vdash}$	-	+			+			_			+	Н	۲
Test Case Workload Model Jussis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A B C D s surviys Seysis & Design A Chiletal Reviewer Infrare Architect Reviewer Infrare Architect Reviewer Infrare Architect Decument Seign Model Improve Test Assets Deployment manager Improve Test Assets Deployment Manage Iteration Evaluate Project Scope and Risk Develop Software Development Plan Closs-out Project Beylow Record Training Material Training Material Training Material Test Results Deployment Material Test Designer Test		ш	_				\perp							ш	L
workload Model Issues List		ш													L
Issues List	efine the System Definition					Test Case					Product Acceptar	nce Plan			Γ
Issues List	anage Changing Requirements	П				Workload Model					Problem Resoluti	on Plan		П	Γ
Activities Define Evaluation Mission Validate Reviewer Validate Build Stability Test and Evaluate Achieve Acceptable Mission Validate Build Stability Test and Evaluate Achieve Acceptable Mission Improve Test Assets Deployment As C D Roles Deployment As C D Roles Deployment As C D Roles Deployment Manager Develop Software Development Plan Close-out Project		A	в с	D		Issues List	\top				Risk Managemer	nt Plan		П	Γ
Define Evaluation Mission Verify Test Approach Veri					Δι		+						+		Г
Verify Test Approach Validate Build Stability Test and Evaluate Achieve Acceptable Mission Improve Test Assets Improve Test Assets Deployment A B C D Roles Deployment Manage Iteration Sales Hareface Specification Sales		П					+						+	Н	۲
Validate Build Stability Tast and Evaluate Achieve Acceptable Mission Improve Test Assets Deployment A B C D Roles Deployment manager Implementor Technical Virter Course developer Artefacts Set Class Sist Class Interface Specification Selign Package Selign Sub-system Selign Sub-system Selign Class Selign Class Interface Selign Class Interface Deployment Infrastructure Product Product Product Product Artwork Deployment Unit Deployment Class Itiles Deployment Unit Develop Spindelines Deployment Unit Develop Spindelines Deployment Unit Develop Spindelines Deployment Unit Develop Spindelines Design Class Design Class Deployment Unit Develop Spindelines Design Class Design Class Deployment Unit Develop Spindelines Development-organisation Assessment Activities Development Unit Develop Spindelines Design Guidelines Design Guidelines Development Class Development Unit Development Unit Development Development Development Unit Development Develo		Н	-				+					-4	+	Н	۲
Test and Evaluate Achieve Acceptable Mission Improve Test Assets Deployment A B C D Roles Deployment A B C D Roles Deployment Manager Implementor Technical Writer Graphic artist Course developer Artefacts Deployment Plan Sign Bodel Sign Sub-system Sign Guidelines Sign Guidelines Sign Guidelines Sign Caluss Sign Calus Sign Sub-system Sign Sub-system Sign Sub-system Sign Calus Sign Sub-system Sign Calus Sign Sub-system Sign Sub-system Sign Sub-system Sign Sub-system Sign Sub-system Sign Calus Sign Sub-system Sub-system Sign Sub-system		ш	_				_					ent		ш	L
Achieve Acceptable Mission Improve Test Assets Deployment A B C D Roles Deployment manager Implementor Technical Writer Graphic artist Course developer Artefacts Deployment Plan Bill of Materials Test Results Deployment Material Test Results Test Results Deployment Infrastructure Product Artwork Deployment Unit Deployment Unit Develops Development Case Deployment Unit Develops Development Development Development Development Development Dev	egrator	ш				Validate Build Stability					Issues List				L
Improve Test Assets Deloyment A B C D Roles Deployment manager Devlop Software Development Plan Close-out Project Deployment Roles Environment Roles Environment Roles Develop Software Development Plan Close-out Project Devlops Segulation Devlops Segulation Devlops Segulation Deployment plan Devlops Material Devlops Segulation Deployment Infrastructure Product Artwork Deployment Unit Devlop Support Material Develop Support Material Development-organisation Assessmer Project-specific Templates Development-organisation Assessmer Project-specific Templates Development-organisation Assessmer Project-specific Templates Development Clase Devlopment Clase De	atabase Designer					Test and Evaluate					Iteration assessn	nent			Γ
Plan for next Iteration Manage Iteration	ftware Architect					Achieve Acceptable Mission	Т				Activities				-
Plan for next Iteration Manage Iteration	oftware Designer	П	_	П		Improve Test Assets	\top			_	Conceive New P	roiect			Г
Roles Roles Roles Roles Deployment manager Implementor Technical Writer Graphic artist Course developer Sist Class Interface Specification Bill of Materials Training Material End-user support Material Test Results Deployment Infrastructure Product Artwork Deployment Unit Develop Software Development Plan Closs-out Project Environment Roles Roles Roles Arefacts Process Engineer Tools Specialist Business Process Analyst Test Designer Test Designer Technical Writer Closs-out Project Environment Roles Process Engineer Tools Specialist Business Process Analyst Test Designer Technical Writer Software Architect Requirement Analyst System Administrator Process Engineer Technical Writer Software Architect Requirement Analyst System Administrator Process Engineer Artefacts Deployment Unit Develop Support Material Develop Support Material Develop Deployment Unit Develop Support Material Development-organisation Assessment Project-specific Templates Development-organisation Assessment Project-specific Templates Development-organisation Assessment Project-specific Templates Development Class Development Unit Develop Support Material Develop Support Material Develop Support Material Develop Support Material Development-organisation Assessment Project-specific Templates Development-organisation Assessment Project-specific Templates Development Case Development Case Development Case Development Case Development Case Development Case Design Guidelines							-	닏	$\overline{}$	_			+	Н	۲
Deployment manager Deployment manager Implementor Technical Writer Graphic artist Course developer Artefacts Deployment Plan Bill of Materials Training Material End-user support Material Test Results Change Request Deployment Infrastructure Product Artwork Deployment Unit Activities Plan Deployment Unit Develop Support Material Development Plan Deployment Plan Deployment Plan Bill of Materials Training Material Test Results Change Request Deployment Infrastructure Product Artwork Deployment Unit Activities Plan Deployment Develop Support Material Develop Support Material Development Class Development Class Development Unit Development Class Development Unit Development Unit Development Development Unit Development Development Unit Development Class Development Unit Development Unit Development Development Class Development Unit Development Unit Development Development Unit Development Development Unit Development Development Class Development Unit Development Development Unit Development Development Unit Development Development Class								-	U				+		F
implementor sk List sion sign Package set Class set Interface Specification sign Package set Class set Interface Specification sign Package set Class set Interface Specification set Interface Specif		ш			Ro								\perp		
Technical Writer Graphic artist Course developer Artefacts Deployment Plan Bill of Materials Training Material Test Results Deployment Material Test Results Deployment Infrastructure Product Product Artwork Deployment Unit Development Class Deployment Unit Development Class Deployment Unit Development Class Deployment Infrastructure Deployment Infrastructure Deployment Unit Develop Support Material Development Class Deployment Unit Develop Support Material Development Deployment Deployment Deployment Unit Develop Support Material Development Class Deployment Unit Develop Support Material Development Class Deployment Unit Develop Support Material Develop Support Material Development Class Deployment Unit Develop Support Material Develop Support Material Development Class Deployment Unit Develop Support Material Develop Support Material Develop Support Material Develop Support Material Development Class Development Cl		ш					\perp	Ш					$oldsymbol{ol}}}}}}}}}}}}}}}}}}$		Ĺ
Graphic artist Course developer Artefacts Deployment Plan Bill of Materials Training Material Trest Results Change Request Perfoct Results Deployment Plan Bill of Materials Training Material Trest Results Change Request Perfoct Results Deployment Infrastructure Product Artwork Deployment Unit Deployment Unit Develop Support Material Develop Support Material Development Class Despondent Class Design Guidelines Design Guid	se-Case Realization	டர				Implementor	┸┚	LĪ			Develop Software	e Development Plan	L		ĺ
Course developer st Class st Class st Interface Specification view Record lange Request lange Request lange Request lend Loser support Material lend Loser support Materia				П		Technical Writer	П	П		٦	Close-out Project		П	П	ĺ
sign Package st Class st Interface Specification view Record ange Request psule psule psule product product Artwork psign Class sign Srecess Analyst sign Srecialist sign Srec	sk List	_		П		Graphic artist	Т	П		_	Environment		Α	В	(
Artefacts Deployment Plan Bill of Materials Business Process Analyst Test Designer Tools Specialist Business Process Analyst Test Designer Test Results Change Request Product Product Product Artwork Deployment Unit Develop Support Material Develop Request Development-organisation Assessmen Project-specific Templates Development-Case Tools Business Product Development-organisation Assessmen Project-specific Templates Development Unit Deployment Unit Deployment Unit Develop Support Material Manage Acceptance Test (Dev site) Product Product Product Deployment Unit Development-organisation Assessmen Project-specific Templates Development-organisation Assessmen Project-specific Templates Development-organisation Assessmen Project-specific Templates Development Unit Development Uni				\blacksquare			т	Н							1
Deployment Plan Bill of Materials Training Material Train Deployment Training Material Train Deployment Training Material Training Material Training Material Training Material Training Material Training Material Train Designer Training Material Train Designer Training Material Train Designer Training Material Train Designer Training Material Training Material Train Designer Training Material Training Material Train Designer Training Material Train Designer Training Material Training Material Train Designer Training Material Training Mate	ion	H			Δ-						5 5 .	r	П	$\overline{}$	г
wiew Record ange Request psule totocol fracin Fast Results Change Requests Deployment Infrastructure pplementary Specifications tal Model tal Model tal Model tal Model Tesa Change Request Deployment Unit Develop Support Material Develop Support Material Develop Support Material Develop Support Material Development Class Desployment Unit Develop Support Material Develop Support Material Development Class Desployment Unit Develop Support Material Development Class Development Unit Develop Support Material Development Class Develop	sign Package	H		\mathbf{H}	Al							•		Щ	۲
Training Material Training Material Training Material Training Material Training Material Training Material Test Designer Tecnical Writer Tecnical Writer Software Architect Requirement Analyst System Administrator Product Product Artwork Deployment Infrastructure Product Artwork Deployment Unit Activities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Product Synthesis alyze Behaviour difine the Architecture sign Components Activities Product Artwork Develop Support Material Manage Acceptance Test (Dev site) Product Deployment Unit Manage Acceptance Test (Inst site) Package Product Provide Access to Download Site Prepare Environment for Project	sion sign Package st Class	H		Ħ			-				I oors specialist				Ĺ
End-user support Material Test Results Change Requests Deployment Infrastructure Product Artwork Deployment Unit Development Class Iffine a Candidate Architecture Inform Architectual Synthesis alyze Behaviour Iffine the Architecture Isign Change Requests Deployment Infrastructure Product Artwork Deployment Unit Development Unit Development Unit Develop Support Material Development Unit Development Unit Development Class Development Unit Develop Support Material Development Clase Tools Business Modelling Guidelines Design Guidelines Design Guidelines Design Guidelines Design Guidelines Design Guidelines Test Guidelines Tools Busines Modelling Guidelines Design Guidelines Design Guidelines Test Guidelines Tools Business Modelling Guidelines Design Guidelines Test Guidelines	sion sign Package st Class st Interface Specification							Ш			n		-		
Test Results Change Requests Deployment Infrastructure Product Product Activities Plan Deployment Develop Support Material Develop Support Materia	sion sign Package st Class st Interface Specification view Record					Bill of Materials						s Analyst		Ш	L
Test Results Change Requests Deployment Infrastructure Product Product Product Product Product Product Product Product Product Process Engineer Activities Plan Deployment Develop Support Material Desployment Unit Develop Support Material Develop	ion sign Package st Class st Interface Specification view Record					Bill of Materials	F					s Analyst			t
Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit Deployment Unit Deployment Class Deployment Unit Deployment Unit Develop Support Material Manage Acceptance Test (Dev site) Product Design Guidelines Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Design Guidelines Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Design Guidelines Design Guidelines Manage Acceptance Test (Inst site) Produce Deployment Design Guidelines Design Guidelines Manage Acceptance Test (Inst site) Product Activities Provide Access to Download Site Propare Environment for Project	sion ssign Package st Class st Interface Specification view Record ange Request					Bill of Materials Training Material					Test Designer	s Analyst			ļ
Deployment Infrastructure Product Produc	ion sign Package st Class st Interface Specification view Record ange Request psule					Bill of Materials Training Material End-user support Material					Test Designer Tecnical Writer	,			
Product Product Artwork Deployment Unit Development-organisation Assessmer Project Sengineer Artefacts Deployment Unit Deployment Unit Development-organisation Assessmer Project Sengineer Artefacts Development Unit Development-organisation Assessmer Project Sengineer Artefacts Development Unit Development Development Development Case Plan Deployment Develop Support Material Develop Support Material Develop Support Material Development Case Tools Manage Acceptance Test (Dev site) Business Modelling Guidelines Design Guidelines Design Guidelines Manual Styleguide Programming Guidelines Test Guidelines Design Guidelines Design Guidelines Design Guidelines Programming Guidelines Test Guideline	sion sign Package st Class st Interface Specification wiew Record ange Request psule					Bill of Materials Training Material End-user support Material Test Results					Test Designer Tecnical Writer Software Architer	et			
Product Artwork Deployment Unit Activities Plan Deployment Development Unit Develop Unit Development Unit Development Unit Development Unit Develop Un	sion sign Package st Class st Interface Specification view Record ange Request psule otocol erface					Bill of Materials Training Material End-user support Material Test Results Change Requests					Test Designer Tecnical Writer Software Architer Requirement Ana	ct			
Deployment Unit Activities Plan Deployment Development-organisation Assessment Project-specific Templates Development Class tites Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Manage Acceptance Test (Dev site) Produce Deployment Unit Manage Acceptance Test (Inst site) Produce Deployment Unit Manage Acceptance Test (Inst site) Product Manage Acceptance Test (Inst site) Programming Guidelines Test Guidelines Test Guidelines Provide Access to Download Site Prepare Environment for Project	sign Package st Class st Interface Specification view Record ange Request psule botocol erface sign Sub-system					Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure					Test Designer Tecnical Writer Software Architer Requirement Ana System Administ	ct sllyst rator			
Deployment Unit Activities Plan Deployment Development-organisation Assessment Project-specific Templates Development Class Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Design Guidelines Design Guidelines Design Guidelines Manage Acceptance Test (Inst site) Package Product Package Product Provide Access to Download Site Propare Environment for Project	sion sign Package st Class st Interface Specification view Record sange Request psule otocol erface sign Sub-system					Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure					Test Designer Tecnical Writer Software Architer Requirement Ana System Administ	ct sllyst rator			
ta Model mponent Class tites Plan Deployment Develop Support Material Development Case Tools Business Modelling Guidelines Design Guidelines Design Guidelines Design Guidelines Manual Styleguide Programming Guidelines Test Guidelines Test Guidelines Test Guidelines Activities Prepare Environment for Project	sign Package st Class st Interface Specification view Record ange Request psule btocol erface sign Sub-system sign Guidelines					Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product					Test Designer Tecnical Writer Software Architer Requirement Ana System Administ Process Enginee	ct sllyst rator			
Plan Deployment Class ties Develop Support Material Develop Support Material Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site) Brogramming Guidelines Manual Styleguide Programming Guidelines Test Guidelines	sion Sign Package st Class st Class st Interface Specification view Record ange Request psule stocol erface sign Sub-system sign Guidelines pplementary Specifications					Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork					Test Designer Tecnical Writer Software Architer Requirement Ana System Administ Process Enginee Artefacts	ct alyst rator			
Develop Support Material Tools	sign Package st Class st Interface Specification view Record ange Request psule stocol erface sign Sub-system sign Guidelines pplementary Specifications sign Class				Δα	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit					Test Designer Tecnical Writer Software Architer Requirement Ana System Administ Process Enginee Artefacts Development-org	at allyst rator r anisation Assessment			
fine a Candidate Architecture from Architectual Synthesis alyze Behaviour fine the Architecture sign Components sign the Database Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site) Programming Guidelines Programming Guidelines Test Guidelines Manual Styleguide Programming Guidelines Test Guidelines Test Guidelines Activities Prepare Environment for Project	sign Package st Class st Interface Specification view Record ange Request psule totocol erface sign Sub-system sign Guidelines pplementary Specifications sign Class ta Model				Ac	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit tivities					Test Designer Tecnical Writer Software Archite Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T	ct plyst rator r r ganisation Assessment femplates			
rform Architectual Synthesis alyze Behaviour fine the Architecture sign Components sign the Database Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site) Package Product Provide Access to Download Site Provide Access to Download Site Prepare Environment for Project	sign Package st Class st Interface Specification view Record ange Request psule blocol erface sign Sub-system sign Guidelines pplementary Specifications sign Class ta Model mponent Class				Ac	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit tivities Plan Deployment					Test Designer Tecnical Writer Software Architet Requirement Ana System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca	ct plyst rator r r ganisation Assessment femplates			
Alyze Behaviour Beta Test Product Manage Acceptance Test (Inst site) Package Product Package Product Provide Access to Download Site Prepare Environment for Project Manual Styleguide Programming Guidelines Test Guidelines Activities Prepare Environment for Project	sion sign Package st Class st Class st Interface Specification view Record sange Request psule otocol erface sign Sub-system sisting Guidelines pplementary Specifications sisting Class ta Model model model sign Sub-system sisting Class standed model model sign Sub-system sisting Class standed model model sign Sub-system sisting Class standed model sign Sub-system sign Class standed sign Sub-system sign Class standard sign Sub-system sign Class standard sign Sub-system sign Class standard sign Sub-system sign				Ac	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit tivities Plan Deployment Develop Support Material					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools	ct tallyst rator r anisation Assessment emplates			
Alyze Behaviour Beta Test Product Manage Acceptance Test (Inst site) Package Product Package Product Provide Access to Download Site Prepare Environment for Project Manual Styleguide Programming Guidelines Test Guidelines Activities Prepare Environment for Project	sion sign Package st Class St Class st Class st Class st Interface Specification view Record sange Request inspect of the Class State of the Class				Ac	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit tivities Plan Deployment Develop Support Material					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools	ct tallyst rator r anisation Assessment emplates			
Manage Acceptance Test (Inst site) Programming Guidelines Test Guidelines Provide Access to Download Site Provide Access to Download Site Propare Environment for Project	sion sign Package sist Class st Class st Interface Specification eview Record lange Request lapsule otocol lerface lesign Sub-system lesign Guidelines lapplementary Specifications late Model lapplement Class late Model lapplement Class lities lefine a Candidate Architecture				Ac	Bill of Materials Training Material End-user support Material Endsuser support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit tivities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site)					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools Business Modelli	ct tilyst rator r anisation Assessment emplates se			
Package Product Provide Access to Download Site Prepare Environment for Project Provide Access to Download Site	sion sign Package sist Class st Interface Specification series Request spage of the sign o				Ac	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit tivities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools Business Modelli Design Guideline	ct allyst retor r r annisation Assessment emplates see			
Provide Access to Download Site Activities Prepare Environment for Project	sion Package sign Package sist Class st Interface Specification sview Record lange Request appsule otocol erface seign Sub-system ssign Guidelines applementary Specifications seign Fub Specifications seign Class tata Model Imponent Class titles serine a Candidate Architecture erform Architectual Synthesis setyze Behaviour				Ac	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Artwork Deployment Unit tivities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools Business Modelli Design Guideline Manual Styleguid	ct dilyst cator r annisation Assessment emplates see e guidelines s e e			
Prepare Environment for Project	sion Package sistin Package sist Class st Interface Specification eview Record lange Request apsule otocol erface sistin Sub-system sistin Guidelines applementary Specifications sistin Class tata Model imponent Class ities effine a Candidate Architecture erform Architectual Synthesis saltyze Behaviour effine the Architecture				Ao	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit tivities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site)					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools Business Modelli Design Guideline Manual Styleguid Programming Gu	ct dilyst cator r annisation Assessment emplates see e guidelines s e e			
	sion Package sign Package sist Class st Interface Specification sview Record hange Request appsule otocol erface sign Sub-system esign Guidelines applementary Specifications sign Guidelines sign Fusion State sign Guidelines sign Guideline				Αα	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit titvities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site) Package Product					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools Business Modelli Design Guideline Manual Styleguid Programming Gu Test Guidelines	ct dilyst cator r annisation Assessment emplates see e guidelines s e e			
Prepare Environment for an Iteration	sion Package sign Package sist Class st Interface Specification sview Record hange Request appsule otocol erface sign Sub-system esign Guidelines applementary Specifications sign Guidelines sign Fusion State sign Guidelines sign Guideline				Ao	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit titvities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site) Package Product					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools Business Modelli Design Guideline Manual Styleguid Programming Gu Test Guidelines	ct dilyst cator r annisation Assessment emplates see e guidelines s e e			
	sion Package sign Package sist Class st Interface Specification sview Record hange Request appsule otocol erface sign Sub-system esign Guidelines applementary Specifications sign Guidelines sign Fusion State sign Guidelines sign Guideline				Ao	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit titvities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site) Package Product					Test Designer Tecnical Writer Software Architet Requirement Ans System Administ Process Enginee Artefacts Development-org Project-specific T Development Ca Tools Business Modelli Design Guideline Manual Styleguid Programming Gu Test Guidelines Activities	ct tatyst retor r r annisation Assessment emplates se ng Guidelines s e			
Fig. 1. Usage map	sion sign Package sticlass st Interface Specification view Record lange Request pseude otoccol erface sign Sub-system ssign Gub-system ssign Gub-system ssign Guidelines pplementary Specifications ssign Calsa ta Model imponent Class tites fifne a Candidate Architecture rform Architectual Synthesis alyze Behaviour fifne the Architecture ssign Components				Ac	Bill of Materials Training Material End-user support Material Test Results Change Requests Deployment Infrastructure Product Product Artwork Deployment Unit titvities Plan Deployment Develop Support Material Manage Acceptance Test (Dev site) Produce Deployment Unit Beta Test Product Manage Acceptance Test (Inst site) Package Product					Test Designer Tecnical Writer Software Architet Requirement Ana System Administ Process Enginee Artefacts Development-org Project-specific T Development Cal Tools Business Modelli Design Guideline Manual Styleguid Programming Gu Test Guidelines Activities Prepare Environr	ct talyst rator r anisation Assessment emplates se g Guidelines s e idelines			

4.2 Interview round 2: Experiences with using RUP

Five project participants with experience from several projects were interviewed to document positive and negative experiences from their use of RUP as well as any improvement suggestions. Note that these interviews are not related to the interviews in round one.

Some of the respondents had experience with more than one role and project type. The five persons interviewed had the following background:

- Respondent 1: Roles: Developer, project manager and test manager. Project types: Web applications with backend logic.
- Respondent 2: Roles: Developer, project manager and test manager (often combined). Project types: Web applications.
- Respondent 3: Roles: Project manager, requirements manager. Project types: Publication systems, banking systems.
- Respondent 4: Roles: Developer. Project types: Mostly system maintenance.
- Respondent 5: Roles: Key account manager. Project types: Secure systems.

Of the five respondents three defined their RUP knowledge as 'good', one as 'medium' and one as 'little'. Following is a summary of common statements from the interview transcriptions showing which respondents having which statements.

			Interview respondents					
+/-	Nodes from interview coding	1	2	3	4	5		
	The RUP training was good				X			
g	Roles defined by RUP	X	X					
ien	Important to have a supporting process				X			
Positive experience	Used inception and elaboration [with success]				X			
ex	SW maintenance projects uses RUPs guidelines	X						
ive	for transition between phases/milestones							
sit	Templates and role definitions are good	X						
Рс	checklists							
	Want to be better at using RUP	X	X	X	X	X		
	Reasonable division in phases and iterations	X						
	To extensive for small projects	X	X	X	X			
Negative experience	Too document-driven				X			
	Too much focus on just the development					X		
	Missing roles for customer contact prior to and					X		
	past the development project							
	Requires good knowledge [of RUP]					X		
	Missing a common standard of use		X	X				
	Does not fit a software maintenance processes	X						
	Miss adaptation to extreme programming	X						
	We do not evaluate our use of RUP	X						
	Continues with old practice				X			
	We have not changed our practice after RUP was		X					
	introduced							
	I have no progress [as a software professional]	X		X		_		
	Missing follow-up during projects		X	X		_		
	Hard to understand RUP		X					

Table 1. Interviews summary

Note that this list is a collection of all statements found relevant to the use of RUP. Some are clear and can be generalized; others are specific to a single project. The definition of the nodes is based on an interpretation of the interviews (due to the constant comparison method).

Besides this overview of experience using RUP, the respondents also had improvement suggestions:

- RUP should be used in a regular manner through the whole project (avoiding deep focus in only parts of the project)
- Projects must be guided in the use of RUP
- Web-projects need more specialized support than RUP can offer
- Establish a project manager forum (for learning and experience exchange)
- Avoid the use of RUP in the case of software maintenance
- Offer support in using and adapting RUP

6. Discussion and conclusion

Offering RUP out-of-the-box leaves all the responsibility of tuning RUP to each individual project. This may cost both time and resources. Good knowledge on RUP is also needed. As the results from interview round one show, the use of RUP is scattered and deviates partly from the RUP guidelines. Project participants seems to end up using some RUP elements mixed with old practice, not as a consequence of deliberate decisions but as a consequence of low knowledge of RUP and how to adapt it.

The phases (and disciplines) of RUP covers the complete lifecycle of a software development project. However, in a real context, as the interviews show, the customer often has done some part of the job initially following an internal process. This may affect the use of RUP later on in the project.

Looking at the results from interview round two we see that most respondents support the idea (in general) of having a guiding process that includes role descriptions and regulates the work in disciplines, phases and iterations. However, <u>all</u> of the respondents feel that they need to and want to be better at using RUP. The reason for this may be that RUP is extremely comprehensive and that the task of fitting this framework to a project may be overwhelming. We also see that four of the respondents find RUP too comprehensive for small projects. This indicates a definitively need for tailoring of RUP in advance of use in projects. Two respondents also miss a common practice for the use of RUP, also indicating the need of a general tailoring of RUP.

In general, the interview results show that providing RUP just in the form of the full documentation (in this case RUP Online – right out of the box) have negative effects, at least not as good effects as one would believe in advance. It is perceived as

too comprehensive and the users have problems finding the parts that would benefit their project. The consequence may be avoidance of use or even worse, wrong use. Two respondents claim that they have not changed their practice of developing software after RUP became available. This resembles with known acceptance models[5]; the methodology must be perceived as <u>useful</u> (will using RUP enhance the job performance?) and it must be perceived as <u>easy to use</u> (will using RUP require low effort?).

Besides doing a thorough adaptation in advance to increase usefulness and ease of use, projects also need practical guidance throughout the project; two respondents miss this type of support, this is also on the list of improvement suggestions. Introducing guidance and mentoring would both improve the degree of use and the effect of use of RUP as well as it would serve as a experience transfer mechanism.

Conclusion: The basic learning from this case study is that a methodology or framework (such as RUP) can not be provided "as is" without experiencing low/wrong use. The users of the methodology need to keep their focus on doing their job (developing software), not struggling to understand the theory. This is actually what the RUP documentation says, but that many unfortunately forget. Introducing a methodology such as RUP is an investment beyond the license fee. In this case the outcome could have been better if the introduction of RUP was carefully managed and not left as an autonomous effort in each project.

A comment: The learning from this study made the company decide to initiate a RUP adaptation process to provide their employees with better process support. This work is described in [6].

7. Further Research

The research reported in this paper, and also in other papers has put emphasis on the challenges in implementing and tailoring RUP for use in an organization [6-8]. Implementing a process framework like RUP can be looked upon as implementation of a new technology in an organization. It would therefore be of interest to study such implementations in spite of technology acceptance models [5] and investigate the success factors of tailoring and introduction of methodologies.

Acknowledgements

The authors would like to thank the participants from the case company. We would also like to thank the SPIKE-project (funded by the Research Council of Norway) for funding and support.

References

- Krutchen, P., The Rational Unified Process: An Introduction. 2nd ed. 2000: Addison-Wesley. 298.
- 2. Yin, R.K., *Case Study Research Design and Methods*. Applied Social Research Methods Series, ed. D.S. Foster. 1994: SAGE Publications.
- 3. Seaman, C., *Qualitative methods in empirical studies in software engineering.* IEEE Transactions on Software Engineering, 1999. **25**(4): p. 557-572.
- 4. Avison, D., et al., Action Research. Communications of the ACM, 1999. 42(1): p. 94-97.
- 5. Riemenschneider, C.K.H., et al., *Explaining software developer acceptance of methodologies: a comparison of five theoretical models.* Software Engineering, IEEE Transactions on, 2002. **28**(12): p. 1135-1145.
- 6. Westerheim, H. and Hanssen, G.K., The Introduction and Use of a Tailored Unified A Case Study. in 31st EUROMICRO CONFERENCE on Software Engineering and Advanced Applications (SEAA). 2005. Porto, Portugal: IEEE.
- 7. Bergström, S., Råberg, L., *Adopting the Rational Unified Process*. 2004, Addison-Wesley. p. 165-182.
- 8. Hanssen, G.K., et al., Tailoring RUP to a defined project type: A case study. in 6th International Conference on Product Focused Software Process Improvement, PROFES. 2005. Oulo, Finland: Springer.