

**UNIVERSIDADE NOVE DE JULHO**  
**PROGRAMA DE MESTRADO PROFISSIONAL EM ADMINISTRAÇÃO**  
**GESTÃO DE PROJETOS**

**DESENVOLVIMENTO ÁGIL DE SOFTWARE E GERENCIAMENTO DE**  
**PORTFÓLIO DE PROJETOS EM AMBIENTES DINÂMICOS: UM ESTUDO DE**  
**CASO DE UM PROVEDOR DE SOLUÇÕES INTEGRADAS**

**FABRÍCIO GARCIA IMBRIZI**

São Paulo

2014

**NOVE DE JULHO UNIVERSITY  
MASTERS OF BUSINESS ADMINISTRATION  
PROJECT MANAGEMENT**

**AGILE SOFTWARE DEVELOPMENT AND PROJECT PORTFOLIO  
MANAGEMENT IN DYNAMIC ENVIRONMENTS: A CASE STUDY OF AN  
INTEGRATED SOLUTIONS PROVIDER**

**FABRÍCIO GARCIA IMBRIZI**

São Paulo

2014

Fabício Garcia Imbrizi

**DESENVOLVIMENTO ÁGIL DE SOFTWARE E GERENCIAMENTO DE  
PORTFÓLIO DE PROJETOS EM AMBIENTES DINÂMICOS: UM ESTUDO DE  
CASO DE UM PROVEDOR DE SOLUÇÕES INTEGRADAS**

Dissertação apresentada ao Mestrado Profissional em Administração - Gestão de Projetos da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de **Mestre em Administração**.

Orientador: Prof. Dr. Antonio Emerson Maccari

São Paulo

2014

Fabício Garcia Imbrizi

**AGILE SOFTWARE DEVELOPMENT AND PROJECT PORTFOLIO  
MANAGEMENT IN DYNAMIC ENVIRONMENTS: A CASE STUDY OF AN  
INTEGRATED SOLUTIONS PROVIDER**

Thesis submitted to the Master of Business Administration in Project Management, Nove de Julho University - UNINOVE, as a partial requirement for the degree of **Master in Business Administration**.

Advisor: Prof. Dr. Antonio Emerson Maccari

São Paulo

2014

Aos meus pais Fábio e Ione  
Obrigado por me educarem  
Amo vocês

## AGRADECIMENTOS

O desenvolvimento desta dissertação de mestrado não seria possível se não fosse pelo apoio de diversas pessoas que estiveram, direta e indiretamente, envolvidas nesse processo de construção nos últimos dois anos, portanto, gostaria de agradecer sinceramente:

- A Deus e ao meu guia espiritual, sempre disponíveis para me alertar, guiar e suportar;
- Aos professores da banca de qualificação, Prof. Dr. Marcos Piscopo e Prof. Dr. Marcirio Chaves, e de defesa, Prof. Dr. Marcos Piscopo (UNINOVE) e Prof. Dr. Yvan Petit (Université du Québec à Montréal), pelas incomensuráveis dicas que contribuíram substancialmente para este estudo;
- Ao meu orientador Prof. Dr. Emerson Maccari que me concedeu a imensa oportunidade, honra e responsabilidade de tê-lo como mestre, sempre me instigando, e incentivando a ir além de minha zona de conforto;
- Aos professores do mestrado que desafiaram e testaram os nossos limites durante todo o curso, permitindo-nos aprimorar nossas habilidades e modo de pensar;
- Aos professores do módulo internacional em Bentley pela experiência única;
- À turma de mestrado, que tive a honra em fazer parte e construir grandes amizades com seres humanos incríveis, com os quais vivi momentos únicos em sala de aula, nos bares, e em Bentley. Essa turma ficou para a história!
- Aos profissionais administrativos do mestrado, sempre disponíveis para auxiliar;
- À empresa pesquisada, por disponibilizar o tempo de seus profissionais que se mostraram disponíveis para cooperar e serem ouvidos;
- Aos gestores da empresa que trabalho por me apoiarem e terem sido compreensíveis com algumas ausências para me dedicar ao mestrado;
- Ao Prof. Dr. José Luís Braga, pela confiança, indicação e incentivos constantes;
- À mulher mais linda do mundo, Isabella, que suportou meus momentos de estresse e extrema ansiedade, e às luzes importunas da madrugada. Te amo muito!
- À minha filha Lara por saber esperar quando precisei me ausentar ainda mais. Te amo;
- Aos meus irmãos Bruno e Júnior, pela torcida por mais uma conquista;
- Aos amigos Fabiano, Leandro, Alexandre e Marcus, por serem grandes “beadas”;
- À Liliana, pelo apoio na revisão ortográfica;
- Aos outros familiares, amigos, e profissionais que, por um lapso de memória, vier a me esquecer, afinal já estou na “reserva”;

## RESUMO

Provedores de soluções integradas (SI) têm grandes desafios em ambientes dinâmicos, repletos de mudanças, imprevisíveis, e estressantes que impactam no gerenciamento do portfólio de projetos (GPP), portanto, é necessário entender o que pode ser feito para suportá-los de tal modo que eles consigam obter o retorno sob o investimento e atingir os seus objetivos organizacionais. Para lidar com um mercado mais turbulento, rápido, arriscado, incerto e imprevisível, alguns autores sugerem a adoção dos valores, princípios e práticas de metodologias ágeis. Essa pesquisa empírica visa identificar como o desenvolvimento ágil de *software* relaciona-se com o GPP em ambientes dinâmicos. Ela é classificada como qualitativa e exploratória; baseada na revisão da literatura de metodologia ágil, GPP e de incertezas e capacidades dinâmicas; e conduzida por um estudo de caso único, cujas evidências foram colhidas por meio de investigação documental, entrevistas guiadas por um protocolo de estudo de caso, e observação participante. Esse estudo está baseado no contexto organizacional de uma empresa de tecnologia da informação (TI) que é focada em soluções integradas, realiza integrações e customizações de *softwares*, e adota a metodologia ágil Scrum para apoiar seu GPP em ambientes dinâmicos. As descobertas da pesquisa sugerem que o Scrum pode contribuir com o GPP, particularmente quanto à i) realocação de recursos, ii) disseminação do conhecimento, e iii) engajamento da equipe de desenvolvimento. Essa pesquisa busca contribuir com a melhor compreensão sobre a relação entre o desenvolvimento ágil de software e o GPP de um provedor de SI, visto que, há poucos estudos empíricos que conectam esses assuntos.

**Palavras-chave:** desenvolvimento ágil de software, gerenciamento de projetos, gerenciamento de portfólio de projetos, ambientes dinâmicos, soluções integradas.

## **ABSTRACT**

Integrated solutions (IS) providers have big challenges in dynamic, full of changes, unpredictable, and stressful environment that impact the project portfolio management (PPM), therefore it is necessary to understand what can be done to support these companies in such a way they can get appropriate returns from investments and achieve their organizational goals. To deal with a more turbulent, speedy, risky, uncertain, and unpredictable market, some authors suggest the adoption of values, principles, and practices of agile methodologies. This empirical research aims to understand how agile software development contributes to PPM in dynamic environments. This study is classified as exploratory and qualitative; based on literature review on agile methodology, PPM, uncertainty and dynamic capabilities; conducted through a unique case study, which evidences were gathered by documental investigation, interviewees guided by a case study protocol, and participant-observation. This study is based on the organizational context of an information technology (IT) company focused on integrated solutions, integrations and customization of software, and that adopt the agile methodology Scrum to support the PPM in dynamic environments. The findings suggest that Scrum can contribute to PPM, particularly in i) resource reallocation, ii) dissemination of knowledge, and iii) engagement of the development team. This research contributes to a better understanding of the relation of agile development software and PPM in an IS provider, since there are few studies that connect these subjects.

**Keywords:** agile software development, project management, project portfolio management, dynamics environments, integrated solutions.



## TABLE OF CONTENTS

<b>LIST OF FIGURES</b> .....	<b>XIII</b>
<b>LIST OF TABLES</b> .....	<b>XIV</b>
<b>LIST OF ACRONYMS</b> .....	<b>XV</b>
<b>1 INTRODUCTION</b> .....	<b>16</b>
1.1 RESEARCH PROBLEM .....	17
1.2 OBJECTIVES.....	18
1.2.1 Main Objective .....	18
1.2.2 Specific Objectives .....	18
1.3 JUSTIFICATION .....	18
1.4 THESIS STRUCTURE .....	21
<b>2 LITERATURE REVIEW</b> .....	<b>22</b>
2.1 AGILE METHODOLOGY .....	22
2.1.1 Emergence of Agile .....	22
2.1.2 Agility, Problem Domain and Agile Methodology Definition.....	25
2.1.3 Values, Principles and Practices .....	26
2.1.4 Benefits and Limitations of Agile Software Development .....	34
2.1.5 Adoption of Agile Methodologies .....	35
2.1.6 Scrum.....	36
2.2 PROJECT PORTFOLIO MANAGEMENT .....	39
2.2.1 Project, Program and Portfolio Definition.....	39
2.2.2 Project, Program and Portfolio Management Definition.....	40
2.2.3 Issues Addressed in Project Portfolio Management .....	42
2.2.4 Value Maximization, Balance and Strategic Alignment .....	43

2.2.5	Key Elements of Project Portfolio Management .....	44
2.2.6	Impacts Adopting Project Portfolio Management .....	45
2.2.7	Project Portfolio Management Processes .....	45
2.2.8	No Standard Approach for Project Portfolio Management .....	47
2.2.9	Portfolio Management and Agile Software Development .....	48
2.3	<b>UNCERTAINTY AND DYNAMIC CAPABILITIES .....</b>	<b>49</b>
2.3.1	Uncertainty .....	49
2.3.2	Dynamic Capabilities .....	50
2.3.3	Conceptual Framework.....	50
2.3.4	Types of Uncertainty .....	51
2.3.5	Project Complexity and Uncertainty .....	52
2.4	<b>SUMMARISING.....</b>	<b>55</b>
<b>3</b>	<b>METHODOLOGY .....</b>	<b>56</b>
3.1	<b>RESEARCH STRATEGY .....</b>	<b>57</b>
3.1.1	Research Strategy Classification .....	57
3.1.2	Type of Research Strategy.....	58
3.1.3	Overview of the Research Process .....	61
3.2	<b>FORMULATING THEORETICAL PROPOSITIONS AND CONSTRUCTING CONSTRUCT .....</b>	<b>62</b>
3.2.1	Formulating Propositions .....	63
3.2.2	Constructing the Construct .....	65
3.3	<b>CASE SELECTION .....</b>	<b>67</b>
3.3.1	Single-Case Design .....	67
3.3.2	Establishing Criteria for Case Selection.....	68
3.3.3	Case Selected.....	68

3.4	DEVELOPING A CASE STUDY PROTOCOL .....	69
3.5	COLLECTING EVIDENCES .....	69
3.5.1	Use of Multiple Source of Evidence.....	70
3.5.2	Use of a Case Study Database .....	72
3.5.3	Use of a Chain of Evidence .....	72
3.6	ANALYSING THE CASE STUDY EVIDENCE.....	73
3.7	REPORTING THE RESULTS.....	74
3.8	CRITERIA FOR JUDGING THE QUALITY OF RESEARCH DESIGNS .....	74
3.8.1	Validity .....	75
3.8.2	Reliability .....	76
<b>4</b>	<b>ANALYSIS AND DISCUSSION.....</b>	<b>77</b>
4.1	ORGANIZATIONAL CONTEXT.....	77
4.1.1	Organizational Structure.....	77
4.1.2	Responsibilities, Challenges, and Limitations of the Project Managers .....	79
4.1.3	Description of the Software Solution .....	80
4.1.4	Unit of Analysis.....	80
4.2	AGILE SOFTWARE DEVELOPMENT PRACTICES.....	82
4.2.1	Organization and Interactions of People .....	85
4.2.2	Software Releases and Assessment .....	86
4.2.3	Customer Collaboration and Development Team Communication .....	90
4.2.4	Planning and Adapting to Changes .....	92
4.3	PPM PRACTICES.....	93
4.3.1	Portfolio of Projects.....	93
4.3.2	PPM Processes.....	94
4.3.3	Resources Allocation.....	97

4.4	TYPES OF UNCERTAINTIES AND PRACTICES TO HANDLE THEM.....	98
4.4.1	Types and Sources of Uncertainties .....	98
4.4.2	Practices to Manage and Control Uncertainty .....	99
4.4.3	Reallocation and Re-optimization of Resources and Capabilities .....	99
4.5	SUMMARISING.....	101
<b>5</b>	<b>FINAL REMARKS .....</b>	<b>102</b>
5.1	CONCLUSION .....	102
5.2	CONTRIBUTION .....	105
5.2.1	For Academia .....	105
5.2.2	For Practice.....	105
5.3	LIMITATIONS .....	106
5.4	FUTURE WORKS .....	106
	<b>REFERENCES .....</b>	<b>107</b>
	<b>APPENDIX A – CASE STUDY PROTOCOL .....</b>	<b>111</b>
	<b>APPENDIX B - DEVELOPMENT TEAM MEMBER OF THE PRODUCT 1.....</b>	<b>114</b>
	<b>APPENDIX C – SPRINT REPORTING TEMPLATE .....</b>	<b>115</b>

## LIST OF FIGURES

Figure 1: The Number of Papers Identified in the Systematic Review of Agile Development Studies per Year .....	19
Figure 2: Number of Occurrences for the Terms Searched by Electronic Database Journals ..	20
Figure 3: Iteration Process Model .....	31
Figure 4: Three Steps to Continuous Integration .....	33
Figure 5: Agile Adoption Continues To Rise .....	35
Figure 6: Scrum Has Become Very Popular .....	36
Figure 7: Scrum Process Overview .....	38
Figure 8: Portfolio, Program, and Projects Interactions .....	40
Figure 9: Comparative Overview of Projects, Program, and Portfolio Management .....	42
Figure 10: Portfolio Management Process Groups and Knowledge Areas Mapping .....	46
Figure 11: The Conceptual Framework to Study the Management of Uncertainty in Project Portfolios .....	51
Figure 12: Types of Changes, Rate of Change, and Impact on the Four Portfolios Studied ...	52
Figure 13: Houston Matrix Quadrant Assessment .....	55
Figure 14: Overview of the Research Process .....	62
Figure 15: Pillars Defined from the Literature Review .....	63
Figure 16: Organizational Structure of the Banking Business Unit .....	78
Figure 17: Backlog's Planning and Prioritization Process .....	83

## LIST OF TABLES

Table 1: Description of Main Agile Development Methodology.....	23
Table 2: Main Differences between Traditional and Agile Methodologies .....	24
Table 3: Agile Testing Strategy.....	32
Table 4: Interfaces between PPM and Scrum.....	48
Table 5: Differences between Risks, Changes, Deviations, and Unexpected Events .....	49
Table 6: System to Score Project Complexity.....	53
Table 7: System to Score Project Uncertainty.....	54
Table 8: Relevant Situations for Different Research Strategies .....	58
Table 9: Research Construct.....	66
Table 10: Types of Documents to Be Collected.....	70
Table 11: Case Study Tactics for Four Design Tests .....	74
Table 12: Main Characteristics of Projects.....	82
Table 13: Main Roles of the Backlog’s Planning and Prioritization Process.....	84
Table 14: Estimated Time for Each Type of Delivery .....	88
Table 15: Main Types of Uncertainties and their Impacts to the Projects and Portfolio .....	98
Table 16: Values, Principles, and Practices Evidenced and Not Evidenced .....	101

## LIST OF ACRONYMS

ABES	Associação Brasileira das Empresas de Software
ASD	Adaptive Software Development
ASDE	Agile Software Development Ecosystems
BCG	Boston Consulting Group
BU	Business Unit
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
DBA	Database Administrator
DSDM	Dynamic Systems Development Method
DSS	Decision Support Systems
FDD	Feature-Driven Development
GDP	Gross Domestic Product
HR	Human Resource
IS	Integrated Solutions
IT	Information Technology
MEC	Ministry of Education
PMI	Project Management Institute
PMP	Project Management Professional
PPM	Project Portfolio Management
QA	Quality Assurance
R&D	Research and Development
RFP	Request for Proposal
RPM	RPM Package Manager
RUP	Rational Unified Process
SCM	Source Code Management
SIT	System Integration Testing
UAT	User Acceptance Testing
XP	Extreme Programming

## 1 INTRODUCTION

The information technology (IT) field has some organizations that are based on projects to support their business, for instance, software houses, consulting firms, IT infrastructure suppliers, and systems integrators. Integrated solutions (IS) providers combine products and systems with services in order to specify, design, deliver, finance, maintain, support, and operate a system throughout its life cycle to address customer's specific business problems (Brady, Davies & Gann, 2005).

These IS providers have to manage their portfolio of projects to achieve organizational strategies and objectives, where projects are evaluated, selected, removed, canceled, postponed, prioritized, authorized, assessed, and monitored. Limited internal resources are allocated and reallocated to the project activities. Some resources are allocated exclusively to a unique project, and others are shared across multiple projects (Cooper, Edgett & Kleinschmidt, 1997a; Reyck et al., 2005; PMI, 2013b).

The project portfolio management (PPM) is specially challenging when changes in the environment combined with high complexity increase uncertainty. Petit and Hobbs (2010) reviewed the literature on management of uncertainty, and proposed a novel framework to study PPM based on concepts derived from the dynamic capabilities and sensemaking. They found in some studies of dynamic capabilities that to achieve a strategic advantage is necessary to reallocate and reoptimize resources and capabilities to adapt to changing environment.

There are IS providers that develop software, changing their own product and integrating it with third-party software to meet the user requirements according to contractual conditions. To deal with a more turbulent, speedy, risky, uncertain, and unpredictable market, authors suggest the adoption of values, principles, and practices of agile methodologies (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006; Leffingwell, 2007; Fernandez & Fernandez, 2008).

In 2012, Brazil was among the top 10 growth in IT, occupying the 7th position in the global ranking of IT investments. The Brazilian IT market, which includes hardware, software and services, moved 60.2 billion dollars, representing 2.67% of the Brazilian gross domestic product (GDP). Of this, 9.5 billion came from software and 15.5 billion from services. Summing



up these two segments they represent more than 40% of the total IT market (Associação Brasileira das Empresas de Software, ABES, 2013).

Given this uncertain and unpredictable scenario but, at the same time, full of opportunities, it is fundamental that IT companies find ways to manage their project portfolios using practices that provide them with better results. In this way they will optimize the resource allocation and raise productivity, maximize profitability, protect their expertise and deliver quality products achieving their strategic objectives.

## 1.1 RESEARCH PROBLEM

There are few empirical studies of agile methodologies, and/or the impact of them on PPM or vice-versa (Dybå & Dingsøyr, 2008; Ionel, 2009; Petit, 2011; Rautiainen, Shantz & Vähäniitty, 2011). On the one hand, there are some benefits in the adoption of agile software development (Dybå & Dingsøyr, 2008), but on the other hand, it is difficult to introduce agile methods into large and complex projects (Dybå & Dingsøyr, 2008; Kruchten, 2011). At the same time, there are studies suggesting the scaling software agility for large companies (Leffingwell, 2007) and how PPM can support it (Rautiainen, Shantz & Vähäniitty, 2011).

This study is based on the organizational context of an IT company focused on integrated solutions that customizes its own software and integrates it with third-party software to meet their customer needs. This company manages a portfolio of projects in a dynamic environment and is implementing the agile methodology Scrum to support the PPM.

Consequently, the motivation of this research is to identify how agile software development contributes to PPM in this organizational context, considering that these subjects are relatively recent in social science research. The research question is formulated as “*How agile software development contributes to project portfolio management in dynamic environments of an integrated solutions provider?*”

The company in this study has been suffering in a dynamic, unpredictable, and stressful environment that has impacted the results of the project portfolio, therefore it is necessary to understand what can be done to support the company in such a way that it can get appropriate returns from investments and achieve the organizational goals.

## 1.2 OBJECTIVES

### 1.2.1 Main Objective

The main objective of this research is to understand how agile software development contributes to PPM in dynamic environments of an integrated solutions provider.

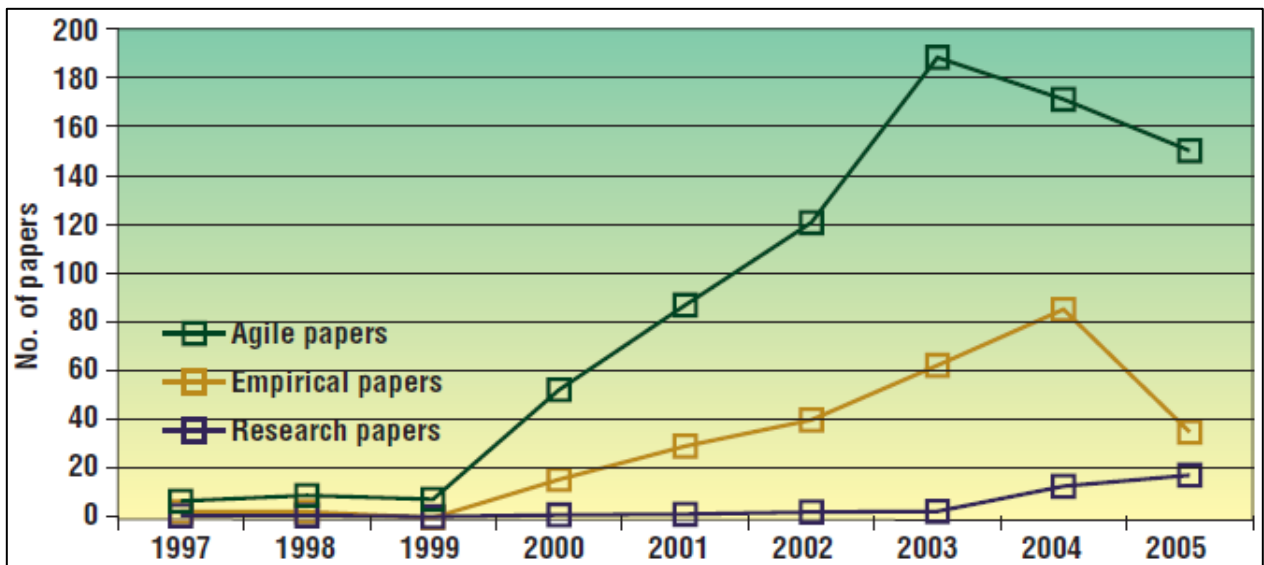
### 1.2.2 Specific Objectives

The specific objectives of this research can be summarized as:

- to identify the common practices in the field of agile software development and PPM adopted by the company;
- to identify the types of uncertainties found in the portfolio studied;
- to identify how the company handles these uncertainties; and
- to make recommendations according to the results of the research and literature review.

## 1.3 JUSTIFICATION

Dybå and Dingsøy (2008) made a systematic review of empirical studies of agile software development. They identified 1996 studies from literature searches, but just 36 were considered research studies of acceptable rigor, credibility, and relevance. Thirty-three of the 36 studies were primary studies, while three were secondary studies. Twenty-five studies investigated extreme programming - XP (76% of 33 studies). They excluded “lessons learned” and “expert opinion-based” papers. Figure 1 illustrates the number of papers identified in the systematic review of agile development studies per year.



**Figure 1: The Number of Papers Identified in the Systematic Review of Agile Development Studies per Year**

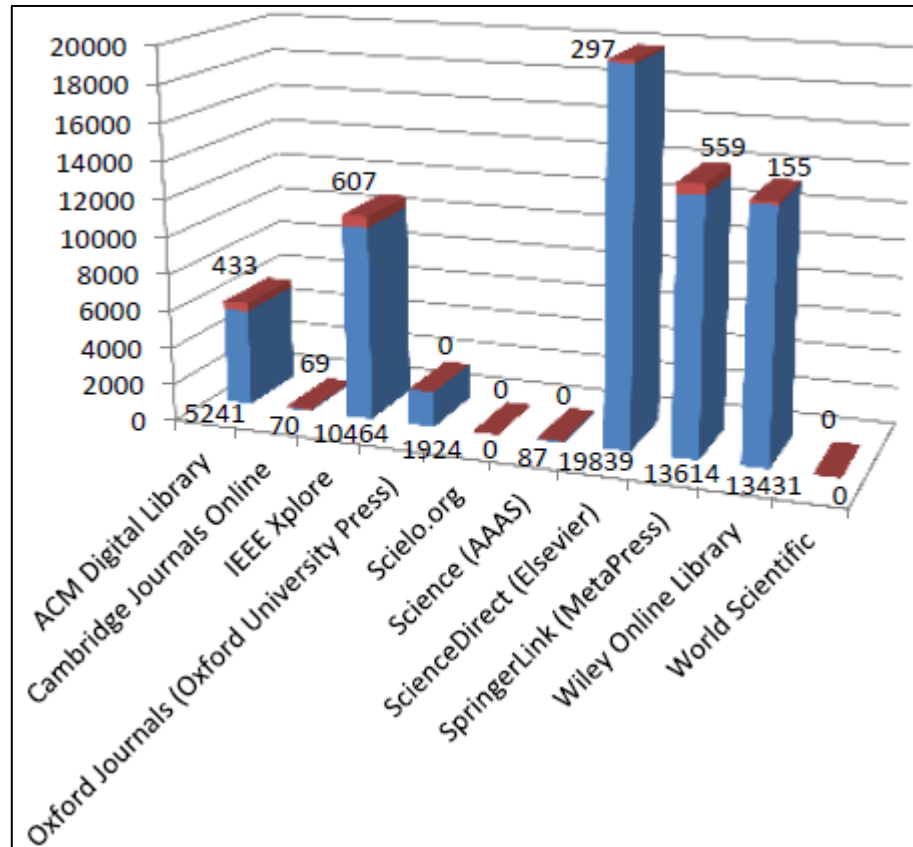
Source: Dybå and Dingsøy (2009).

Figure 1 divided the papers into three groups: all agile papers, the proportion of the agile papers that were empirical, and the proportion of the empirical papers that were research papers. Dybå and Dingsøy (2008) found in their review the need for more empirical studies of agile development methods. Management-oriented agile approaches, such as Scrum, despite their popularity, are clearly the most under-researched. They believe that the current state of theory and research on agile methods is nascent, which suggests a need for exploratory qualitative studies.

According to Ionel (2009), agile methodologies are increasingly used by software development companies, but there are few and relevant papers about it. Most case studies are focused on Scrum and XP. Petit (2011) suggests that there are few studies on the impact of agile development in PPM.

Santos Filho (2012) also made a review of empirical studies using the virtual library *Portal de Periódicos* of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), a Brazilian foundation of the Ministry of Education (MEC) that plays a key role in the expansion and consolidation of post-graduate studies (master and doctorate). This library concentrates articles of the national and international electronic database journals best known worldwide.

Santos Filho (2012) filtered the search considering the computer science knowledge area, 10 databases, and the terms “*information technology portfolio management*” and “*agile software development*”. He found 2120 studies, as depicted in Figure 2, but just 18 studies were considered research studies of acceptable rigor, credibility, and relevance. This result corroborates the necessity of more studies in this field.



**Figure 2: Number of Occurrences for the Terms Searched by Electronic Database Journals**

Source: Santos Filho (2012).

Figure 2 illustrates, for each of the 10 electronic database journals used in the search, the numbers of occurrences for the terms “*information technology portfolio management*” and “*agile software development*” at the top, and the numbers of occurrences for the term “*information technology portfolio management*” at the bottom of each bar.

## 1.4 THESIS STRUCTURE

This study is structured into five chapters, considering this introduction. Second chapter presents a literature review of agile methodology, PPM, uncertainty, and dynamic capabilities.

Chapter three presents the methodological aspects of this research, including the research strategy, formulation of theoretical propositions, construction of construct, the case selection, the development of a case study protocol, how to collect and analyze the evidences, and how to present the results.

Chapter four presents the analysis and discussion of the results, based on collection of evidences and methodological review.

Finally, chapter five presents the conclusion summarizing the findings of the research, the contributions for academia and practice, the limitations of the study, and suggestions for future works.

## 2 LITERATURE REVIEW

This chapter provides a review of the literature about the three theoretical poles that support this research: agile methodology, PPM, and uncertainty and dynamic capabilities.

### 2.1 AGILE METHODOLOGY

According to Highsmith (2002, p. 52), “Methodologies are how teams work together to solve a particular kind of problem ... Methodology forms a framework within which people can work effectively together, but it cannot, and does not, substitute for talent and skill.”. For Cockburn (2006), “Your ‘methodology’ is everything you regularly do to get your software out. It includes who you hire, what you hire them for, how they work together, what they produce, and how they share. It is the combined job descriptions, procedures, and conventions of everyone on your team”.

#### 2.1.1 Emergence of Agile

During the 70’s and until the early 1990s some methodologies were highly used to develop software driven by an extensively documentation, process-oriented, plan-based, and engineering-based, hereby called traditional, rigorous, plan-driven, or waterfall methodologies. It was a time with certain stability and predictability. After the early 1990s the market changed becoming more turbulent, speedy, risky, uncertain, unpredictable, and replete of competitive challenges. In other words, it is a time for faster and different types of changes.

According to Beck et al. (2001), to deal with this new environment, seventeen specialists, experienced software developers and agile practitioners met in February 2001 and created a statement that was an alternative to rigorous methodology. They called it *The Manifesto for Agile Software Development*, hereby called Manifesto. Table 1 presents some agile methodologies created by them.

**Table 1: Description of Main Agile Development Methodology**

<b>Methodology</b>	<b>Description</b>	<b>Authors</b>	<b>Year</b>
<b>Lean Development</b>	An adaptation of principles from lean production and, in particular, the Toyota production system to software development. Consists of seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole.	Bob Charette	1993
<b>Dynamic Systems Development Method (DSDM)</b>	Divides projects in three phases: pre-project, project life-cycle, and post project. Nine principles underlie DSDM: user involvement, empowering the project team, frequent delivery, addressing current business needs, iterative and incremental development, allow for reversing changes, high-level scope being fixed before project starts, testing throughout the lifecycle, and efficient and effective communication	Arie van Bennekum et al.	1994
<b>Scrum</b>	Focuses on project management in situations where it is difficult to plan ahead, with mechanisms for “empirical process control”; where feedback loops constitute the core element. Software is developed by a self-organizing team in increments (called “sprints”), starting with planning and ending with a review. Features to be implemented in the system are registered in a backlog.	Ken Schwaber, Jeff Sutherland and Mike Beedle	1995
<b>Crystal Methods</b>	A family of methods for co-located teams of different sizes and criticality: Clear, Yellow, Orange, Red, Blue. The most agile method, Crystal Clear, focuses on communication in small teams developing software that is not life-critical. Clear development has seven characteristics: frequent delivery, reflective improvement, osmotic communication, personal safety, focus, easy access to expert users, and requirements for the technical environment	Alistair Cockburn	1998
<b>Feature-Driven Development (FDD)</b>	Combines model-driven and agile development with emphasis on initial object model, division of work in features, and iterative design for each feature. Claims to be suitable for the development of critical systems. An iteration of a feature consists of two phases: design and development.	Jeff De Luca and Peter Coad	1999
<b>Extreme Programming (XP)</b>	Focuses on best practice for development. Consists of twelve practices: the planning game, small releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-h week, on-site customers, and coding standards. The revised “XP2” consists of the following “primary practices”: sit together, whole team, informative workspace, energized work, pair programming, stories, weekly cycle, quarterly cycle, slack, 10-minute build, continuous integration, test-first programming, and incremental design. There are also 11 “corollary practices”.	Kent Beck, Ward Cunningham, and Ron Jeffries	1999
<b>Adaptive Software Development (ASD)</b>	The practices are driven by a belief in continuous adaptation. It is based on a Speculate-Collaborate-Learn life cycle that is dedicated to continuous learning and oriented to change, reevaluation, peering into an uncertain future, and intense collaboration among developers, management, and customers.	Jim Highsmith	1999

Source: adapted from Dybå and Dingsøy (2008).

Table 1 presents the authors, the estimated year of release, and the main concepts behind the main agile methodologies. Section 2.1.6 provides further information about the agile methodology for project management, Scrum.

#### 2.1.1.1 Traditional Versus Agile Methodologies

Fernandez and Fernandez (2008, p. 15) concluded that “agile practices, including project management, grew out of a need to manage projects characterized by complexity and uncertainty with responsiveness and adaptability. When goals and solutions are unclear and there is high volatility, there is particular need for alternative approaches to managing projects”. Table 2 illustrates main differences between traditional and agile methodologies described by Dybå and Dingsøy (2008).

**Table 2: Main Differences between Traditional and Agile Methodologies**

	<b>Traditional view</b>	<b>Agile perspective</b>
Design process	Deliberate and formal, linear sequence of steps, separate formulation and implementation, rule-driven	Emergent, iterative and exploratory, knowing and action inseparable, beyond formal rules
Goal	Optimization	Adaptation, flexibility, responsiveness
Problem-solving process	Selection of the best means to accomplish a given end through well-planned, formalized activities	Learning through experimentation and introspection, constantly reframing the problem and its solution
View of the environment	Stable, predictable	Turbulent, difficult to predict
Type of learning	Single-loop/adaptive	Double-loop/generative
Key characteristics	Control and direction Avoids conflict Formalizes innovation Manager in controller Design precedes implementation	Collaboration and communication; integrates different worldviews Embraces conflict and dialectics Encourages exploration and creativity; opportunistic Manager is facilitator Design and implementation are inseparable and evolve iteratively
Rationality	Technical/functional	Substantial
Theoretical and/or philosophical roots	Logical positivism, scientific method	Action learning, John Dewey's pragmatism, phenomenology

Source: adapted from Dybå and Dingsøy (2009).

Table 2 illustrates the traditional view and agile perspective, considering the design process, goal, problem-solving process, view of the environment, type of learning, key characteristics, rationality, and theoretical and/or philosophical roots.



### 2.1.2 Agility, Problem Domain and Agile Methodology Definition

Highsmith (2002) answers three fundamental questions in his book titled *Agile Software Development Ecosystems*: what is agility? What kinds of problems does agility solve best? What are agile software development ecosystems (ASDEs)?

For Highsmith (2002), “Agility is the ability to both create and respond to change in order to profit in a turbulent business environment. Agile organizations harness or embrace change by being better than competitors at responding to changing conditions and by creating change that competitors can’t respond to adequately. Agile organizations are able to change directions quickly and flexible to adopt new ways to do the work... Agility organizations understand that balancing on the edge between order and chaos determines success.” Kruchten (2011) enforces the definition of Highsmith as opposed to defining agility by a labeled set of practices, which seem to suggest just common sense.

Goldman (as cited in Cockburn, 2006) defines agility as “dynamic, context-specific, aggressively change-embracing, and growth-oriented. It is not about improving efficiency, cutting costs, or battening down the business hatches to ride out fearsome competitive ‘storms.’ It is about succeeding and about winning: about succeeding in emerging competitive arenas, and about winning profits, market share, and customers in the very center of the competitive storms many companies now fear”.

According to Highsmith (2002), “problems characterized by change, speed, and turbulence are best solves by agility. Extreme or complex projects – those that have an accelerated time schedule combined with significant risk and uncertainty that generate constant change during the project”.

An ASDE is “a holist environment that includes three interwoven components – a ‘chaordic’ perspective, collaborative values and principles, and a barely sufficient methodology. The focal points of agile development are people, relationships, and uncertainty. Focusing on people and their interactions and giving individuals the power to make quick decisions and to self-adapt their own process are key to agile ecosystems” (Highsmith, 2002).

In a “chaordic” perspective, teams are decentralized, independent, and interact in self-organizing ways, guided by a set of simple rules that deliver results. Values and principles aid define a collaborative culture within which individuals interact with one another as a team

creating an environment where people are comfortable to work. Barely sufficient or streamlined methodologies focus on activities that add value to the customer and eliminate the rest. It is practice-centered instead of process-centered. In a ‘chaordic’ environment, adopting bare sufficient methodology creates the perfect conditions for innovation – the ability to create new knowledge that provides business value – and creativity to flourish (Highsmith, 2002).

### 2.1.3 Values, Principles and Practices

This section describes the values, principles, and practices found on the literature review of agile methodologies.

#### 2.1.3.1 Values

The Manifesto states that:

“We are uncovering better ways to developing software by doing it and helping others do it. Through this work we have come to value:

- (1) **Individuals and interaction** over processes and tools
- (2) **Working software** over comprehensive documentation
- (3) **Customer collaboration** over contract negotiation
- (4) **Responding to change** over following a plan

This is, while there is value in the items on the right, we value the items on the left more” (Beck et al., 2001).

Processes and tools are important, but talented people and the interaction between them are by far more important, therefore processes and tools must be adapted to talented people, not the opposite. Comprehensive documentation is essential, but just the minimum necessary to deliver real, functional and working software that the customers can see working on screen. The working software is the only thing that shows what the development team has achieved, no matter what has been documented (Highsmith, 2002; Cockburn, 2006).

Contract negotiation must have limits within which the parties can move, but only the customer collaboration with the development team during the project execution can deliver the suitable results for the customers’ real needs. Collaboration means close interactions of

individuals, making decisions together. Planning is useful, important and necessary, but adapting to further changes to the plan and understanding the limits of planning is even more important and desirable. Each one of the agile methodologies contains planning activities and mechanisms for dealing with changing priorities (Highsmith, 2002; Cockburn, 2006).

### 2.1.3.2 Principles

Agile methodologies i) focus on the set of problems described in section 2.1.2; ii) are customer driven, which means, getting the customer involved and in control all the time, delivering value first; iii) focus on people considering individual skills, collaboration, exchange, and communication that promote innovation and flexibility; and iv) are practice-driven, not process-driven. Beck et al. (2001) define 12 principles in the Manifesto, as follows:

**(1) Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**

Cockburn (2006) comprehends that delivering working software early and frequently allows for continued wins and early feedback about the requirements, the team, and the process, and allows changes in project direction and priorities.

**(2) Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**

Highsmith (2002) addresses the issues of predictability versus adaptability. For him, in times of uncertainty and turbulence, “Planners ‘presume’ they know more about the future than they actually do, and they are at the same time overly cautious and fail to act quickly enough”. It is better to adapt to changes instead of fighting them. He suggests to embrace changes considering that: i) facilitating will be more important than controlling change; ii) getting better at rework becomes a virtue; iii) change “control” is best focused on the final product components; iv) feedback must be built into every level of development; and v) facilitating change requires multi-level processes.

**(3) Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**

Cockburn (2006) has found most project running in one- to three-month cycles, and shorter cycles are rare, because the customers usually are not prepared to take in more frequent changes than that.

**(4) Business people and developers must work together daily throughout the project.**

Highsmith (2002) understands that the customer must be in control over features and priorities and the relationship between the customer and the development team must be a collaborative partnership. These principles require that both share the responsibility, stimulating constant communication and conversation to understand the customer's real needs. Delivering working software frequently enforces the necessity of interaction between customer and development all the time, on a daily basis. Cockburn (2006) alerts that the longer it takes to get information to and from the developers, the more problems the project will present.

**(5) Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**

According to Highsmith (2002) and Cockburn (2006) the key for success is the support and trust in motivated, valued, and skilled people, in such a way, they can make good decisions based on their experience and knowledge, deciding how to get the job done and getting the job done. Since this is a people-oriented, and not a process-oriented approach, the focus is to capitalize on each individual and team, and tailor processes according to their unique strengths.

**(6) The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**

Highsmith (2002) explains that information exchange between people having direct conversation is more effective to understand the problems and solutions than well formatted documentation and processes. Documentation is necessary to support the interaction between knowledgeable people, but it is only 15 to 20 percent of the understanding required. He stated that "Agile practitioners lean toward interaction, whereas rigorous methodologists lean toward documentation".

**(7) Working software is the primary measure of progress.**

Cockburn (2006) relies on the honesty that comes with running code than on promissory notes in the form of plans and documents. Highsmith (2002) enforces that delivering working software frequently implies delivering value to the customer in a way that customer can sense that the deliverables are evolving.

**(8) Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**

Cockburn (2006) insists that people working long hours are counter-productive during overtime and regular hours too, introducing more errors into the code. This invalidates all efforts of the team working extra hours, beyond keeping them tired and with lower morale.

**(9) Continuous attention to technical excellence and good design enhances agility.**

Highsmith (2002) reinforce the importance of technical excellence to produce high-quality software, which includes multi-level testing; simple, readable, and understandable code; inspection, and refactoring. Cockburn (2006) says that designers have to produce tidy, good, and well-encapsulated designs, review and improve regularly to keep them up-to date to the changing requirements.

**(10) Simplicity – the art of maximizing the amount of work not done is essential.**

Cockburn (2006) and Highsmith (2002) defends that producing a simple design that can face changes effectively is more difficult. At same time, it raises complex and intelligent behavior.

**(11) The best architectures, requirements, and designs emerge from self-organizing teams.**

Highsmith (2002) use five key ideas about complex system to explain the emerging: i) complex systems, be they biological or human, are composed of independent, decentralized agents; ii) these independent agents, in the absence of centralized control, will self-organize; iii) self-organization will create complex behavior and emergent results that are not evident from studying the agents themselves; iv) rich information flows in an ecosystem balanced at the edge of chaos define the most effective pattern for generating emergent results; and v) simple,

generative rules guide the creation of complex behaviors. Cockburn (2006) insist that the architecture have to adjust over time and grows in steps that can follow the changing requirements of the customer and changing knowledge of the development team.

**(12) At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.**

Cockburn (2006) defends the importance to reflect on what the team is doing after each regular interval. It is so important if the team wants to evolve their methodology to be agile, effective, and fitting the project ecosystem.

#### 2.1.3.3 Practices

Leffingwell (2007) reviewed a number of agile methodologies (Scrum, XP, LSD, DSDM, FFD, and one iterative and incremental method, Rational Unified Process (RUP)) and he concluded that all these methods have seven best practices in common:

##### **(1) The Define/Build/Test Component Team**

Leffingwell (2007) describes the *define/build/test component team* as the fractal on which agile development is based. To develop working software in an iteration time box, teams must contain three basic skills: (i) product owners, who work with the customers and stakeholders to define the solution; (ii) team members, who create the code; and (iii) testers, who are responsible for acceptance test. He uses *define/build/test* to illustrate that the process is concurrent, collaborative, and atomic, in other words, he states that “with agile, the organization must be reorganized so that each team has all the skills-product definition, software development, and testing-necessary to define/build/test and deliver each story”.

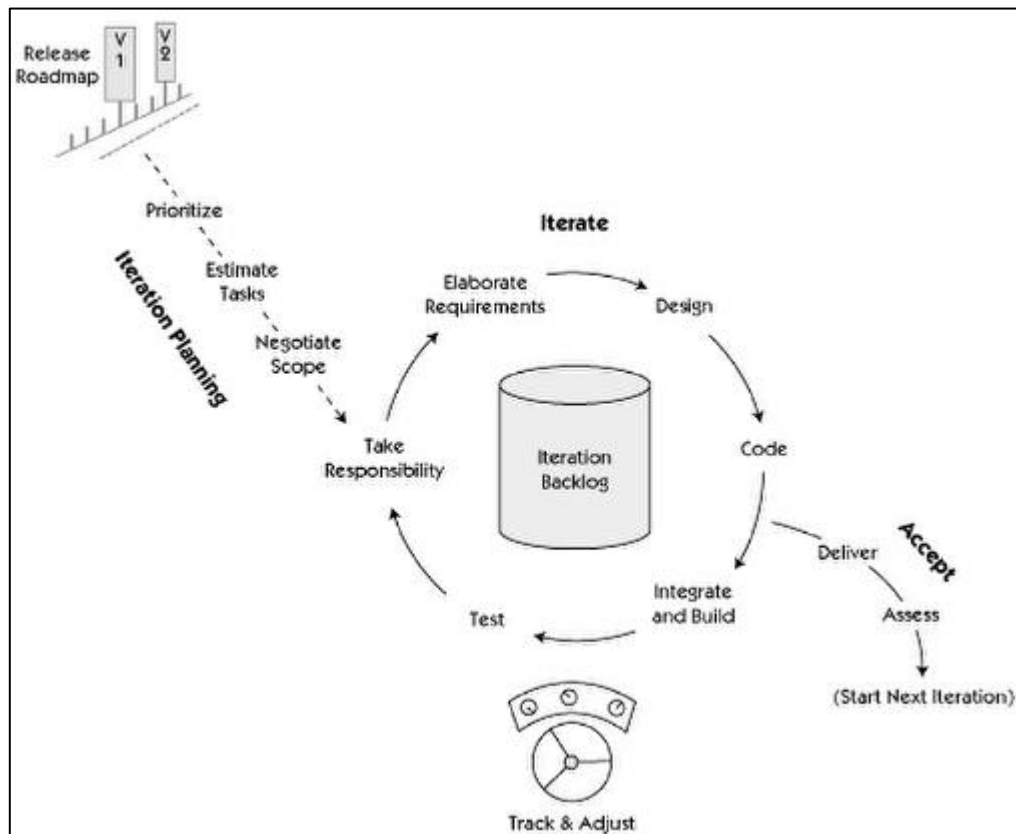
##### **(2) Two-Level Planning**

Leffingwell (2007) verifies two-level planning for agile: (i) release plans, which are intentionally coarse-grained, less comprehensive, and less precise, define the planning for delivering features to the customers in long-range; and (ii) iteration plans, which are precise, confident, define the planning for time-boxed increment of functionality in near term. Each

release consisted of a set of iterations, and both driven by the product backlog, which is defined by product owners.

### (3) Mastering the Iterations

According to Leffingwell (2007), the iteration is the heartbeat of agility, “the ability of the team to create working, tested, value-delivered code in a short time box – with the goal producing an increment of potentially shippable code at the end of each iteration”. Figure 3 shows an iteration process model. He and Krebs (2009) recommend an iteration of two weeks in length.



**Figure 3: Iteration Process Model**

Source: Leffingwell (2007).

Figure 3 illustrates the iteration process model, composed of three phases: (i) iteration planning, during which the product backlog is reviewed, prioritized, estimates are defined, and the work to be done is established; (ii) iterate, which is the development and test of the backlog items; and (iii) accept, which is the delivering and acceptance of the working software.

#### (4) Smaller and More Frequent Releases

Deliver smaller and more frequent releases to customers has some business benefits: rapid response to changing marketing conditions and more frequent feedback from users, reducing the business risk (Leffingwell, 2007). This agile approach delivers a return on investment early because the payback period runs parallel while the software is being developed (Krebs, 2009).

#### (5) Concurrent Testing

Leffingwell (2007) lists four principles that drive agile tests: (i) all code is tested code; (ii) tests are written before, or concurrently with, the code itself; (iii) testing is a team effort. Testers and developers all write tests; and (iv) automation test is the rule, not the exception. He found that most agile teams are focused on four types of agile testing strategy, as described in Table 3.

**Table 3: Agile Testing Strategy**

<b>Unit testing</b>	<b>Acceptance testing</b>	<b>Component testing</b>	<b>System, performance, and reliability testing</b>
Developers write unit tests for every class and every method.	Testers/product owners write functional or acceptance tests for each new user story (requirements).		Automated builds assemble all system components into a daily system build.
Each unit test returns a “pass” or “fail” against the developer’s build.	Acceptance tests are elaborated and written during iteration planning and execution.	Developers and testers write component-level tests.	Unit tests, component, acceptance, and regression tests are run against the daily build.
All unit tests must pass before code is checked in.	Acceptance tests are run during the iteration and serve as acceptance check-points for the iteration's stories.	Component tests are run during the iteration to assure that the "system still runs."	Developers and QA personnel create performance, stress, and load tests to test the boundaries of the system.
Automated unit tests are run frequently (or continuously) against an integrated build of the system.	Acceptance tests are automated wherever possible and are added to the regression test suite at each iteration.	New component tests are linked and automated into the regression test suite.	These tests are run as often as possible, ideally once daily, nominally once or twice during the course of an iteration, worst case in a hardening iteration toward end of the release.

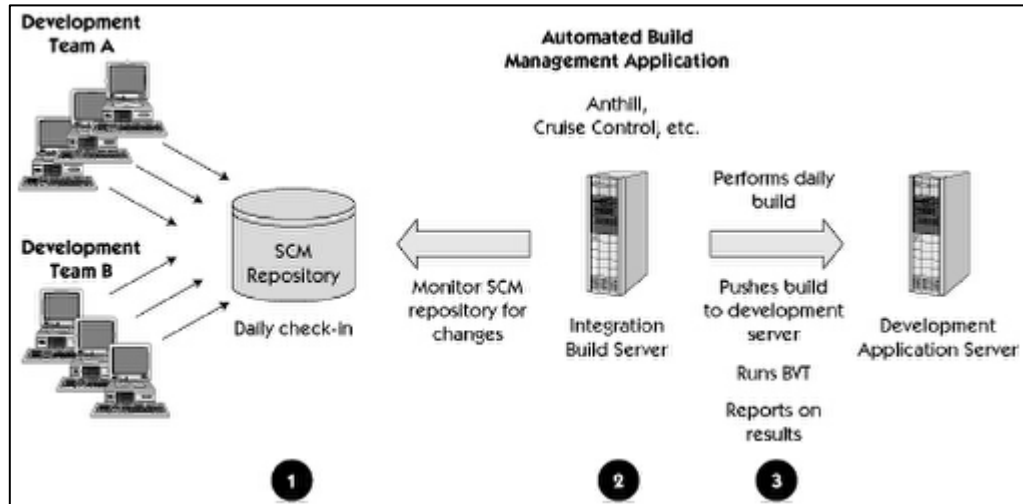
Source: adapted from Leffingwell (2007).

#### (6) Continuous Integration

For Leffingwell (2007), “continuous integration is a process, supported by tools, which results in a working build of the system on at least a daily basis. With continuous integration, the



results of all work are available every day for evaluation and inspection". He described three steps to continuous integration, as showed in Figure 4.



**Figure 4: Three Steps to Continuous Integration**

Source: Leffingwell (2007).

Figure 4 illustrates the three steps: (1) source code integration, which defines that there must be a single source code management repository (SCM repository) and all source code must be checked into SCM repository at least daily; (2) automated build management, which monitor the SCM repository, builds the binary files for these source codes, and report the success or failure of the building process; and (3) automating build verification test, which build management application delivers the binary files to some test servers, which emulate the deployed application.

He lists some benefits of continuous integration, as follows: i) less time is spent hunting bugs caused by one person's code stepping on another person's code, ii) rapid discovery of misunderstandings occurs between (a) developers working on the same component and (b) developers working on different components, iii) defects are discovered while the work is fresh in everyone's minds, and team members are still present to make the corrections efficiently, iv) the compounding effect of undiscovered defect on top of undiscovered defect is greatly mitigated, v) code in the SCM repository is safe and secure, and it reverts to collective (and company)

ownership daily, vi) all new code is compiled every day, and vii) the progress of the system as a whole can be measured daily (Leffingwell, 2007).

### **(7) Regular Reflection and Adaptation**

After each iteration and release, the agile team assesses and improves their processes of work. It is a great opportunity to eliminate obstacles and impediments that prevent the agile team to be more productive, effective, and deliver working software to customer (Leffingwell, 2007).

#### 2.1.4 Benefits and Limitations of Agile Software Development

Dybå and Dingsøy (2008) found some benefits of agile software development: customer collaboration, work processes for handling defects, learning in pair programming, thinking ahead for management, focus on current work for engineers, better estimation, improved job satisfaction, productivity, changes are incorporated more easily and business value is demonstrated more efficiently, and increased customer satisfaction.

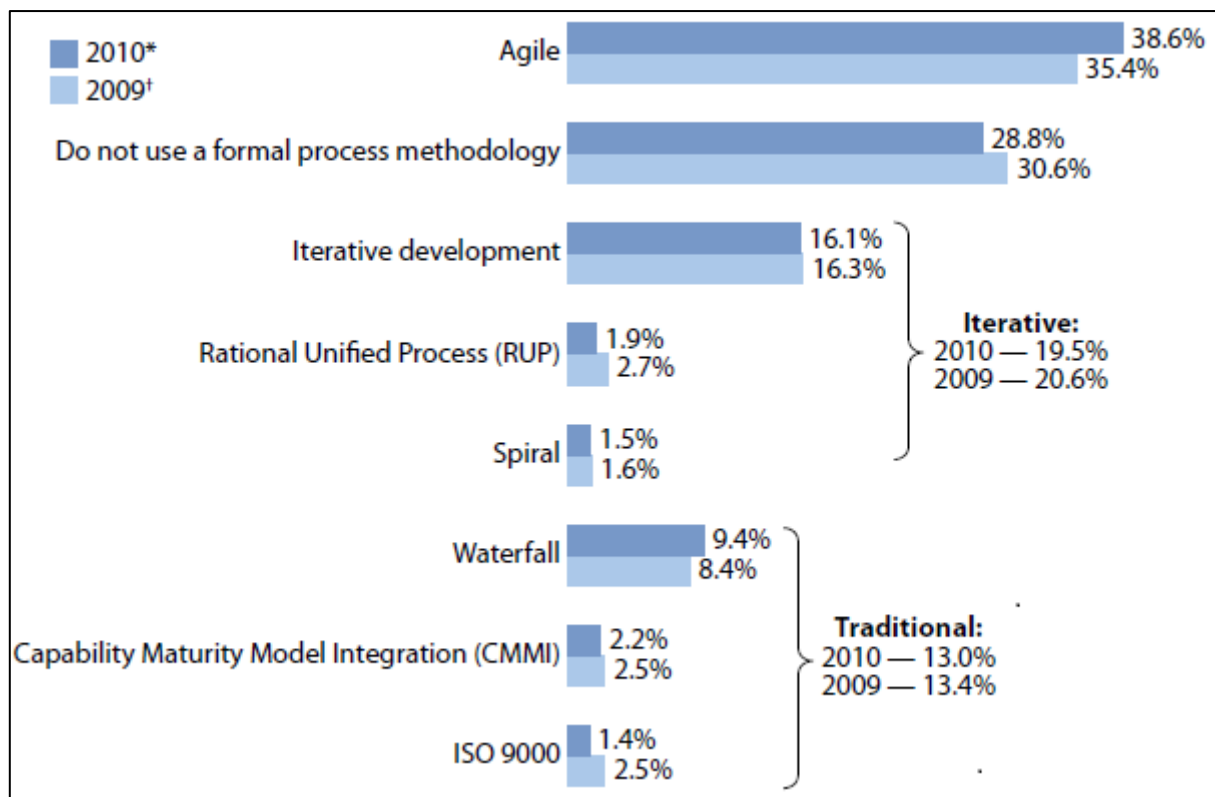
The strongest, and probably most relevant, evidence for practice is from the studies of mature agile teams, which suggests that it is necessary to focus on human and social factors in order to succeed. Specifically, it seems that a high level of individual autonomy must be balanced with a high level of team autonomy and corporate responsibility. It also seems important to staff agile teams with people that have faith in their own abilities combined with good interpersonal skills and trust (Dybå and Dingsøy, 2008).

Dybå and Dingsøy (2008) found some limitations of agile software development: the role of on-site customer seems to be unsustainable for long periods; it is difficult to introduce agile methods into large and complex projects, and team members are less interchangeable in agile teams, having consequences on how projects are managed.

Kruchten (2011) suggests that agile methods may fail when applied in a context different from those they have been created for. In his experience, the contextual factors that may bring concern are: size, large system with a lack of architectural focus, software development not driven by customer demand, lack of support from surrounding stakeholders, traditional governance, novice team, and very high constraints on some quality attributes (safety-critical system, real-time constraints).

### 2.1.5 Adoption of Agile Methodologies

According to West et al. (2010, 2011), “in the past few years, agile processes have not only gained increasing adoption levels; they have also rapidly joined the mainstream of development approaches”. They showed that, in the research conducted by Forrester Research in 2009 and 2010, 35.4% and 38.6%, respectively, of IT professionals stated that agile closely reflects their development processes. On other hand, 34% and 32.5% are still using either an iterative or waterfall development process, and 30.6% and 28.8% do not use a formal process methodology, as depicted in Figure 5.



**Figure 5: Agile Adoption Continues To Rise**

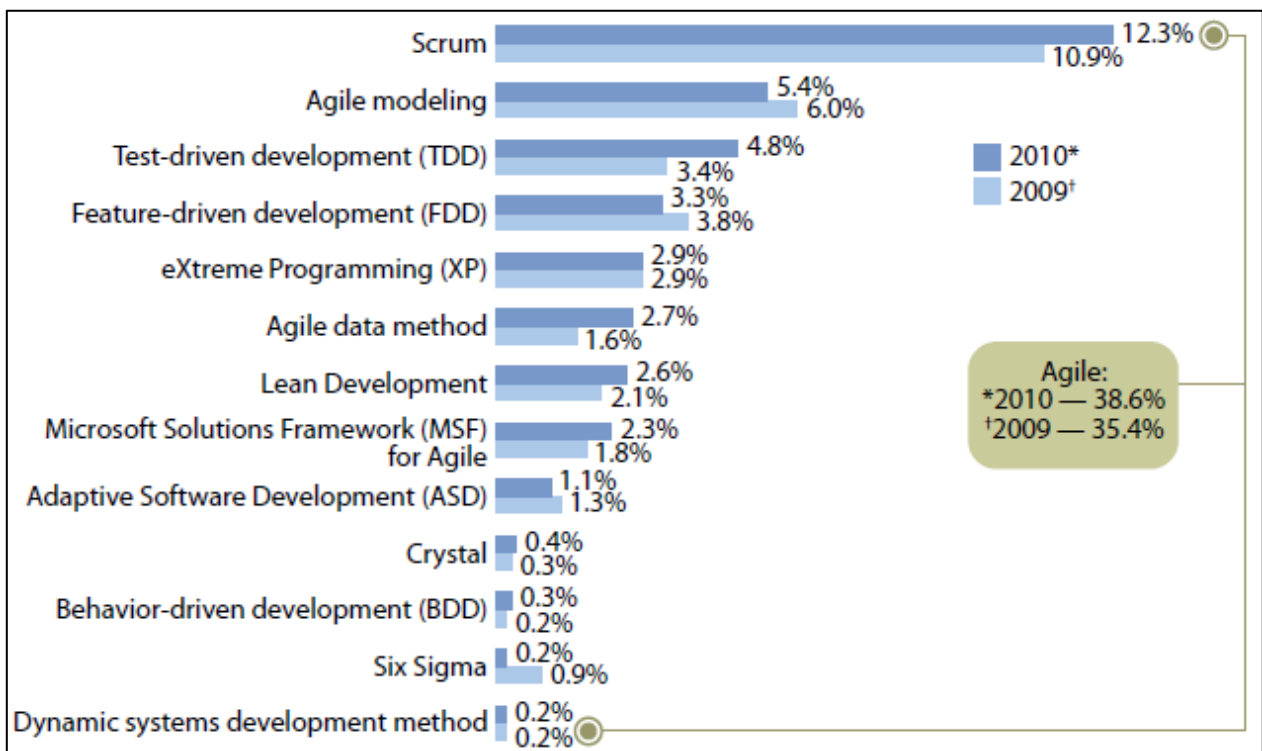
Source: adapted from Forrester Research, Inc. (as cited in West et al., 2011).

Figure 5 illustrates the methodology that most closely reflects the development process the respondent was using at that time. West et al. (2010) identified important differences in agile adoption between technology industry firms – which sell products developed by the development

teams under constant pressure, and IT departments in other types of industries – outputs so developed affect how the organization’s employees, partners, and customers work. In general, technology industry firms adopt more agile practices than IT departments.

### 2.1.6 Scrum

Scrum is the most popular and adopted agile methodology, according to Forrester Research (as cited in West et al., 2011, p. 3). The research showed that 31.9% of the application developers adopting some agile methodology are using Scrum. Figure 6 illustrates the methodology that most closely reflects the development process the respondent was using at that time, considering only the agile methodology defined on Figure 5.



**Figure 6: Scrum Has Become Very Popular**

Source: adapted from Forrester Research, Inc. (as cited in West et al., 2011).

Scrum is a framework to manage and develop complex products, within which it is possible to employ various processes and techniques. Scrum clarifies the relative efficacy of

product management and development practices. It is composed by teams (and their respective roles), events, artifacts, and rules that gear the relationship and interactions between them (Schwaber & Sutherland, 2013).

Scrum is based on empiricism, which maintains that knowledge comes from experiences and decision making based on what is known. Scrum applies an incremental and iterative approach to optimize predictability and risk control. The empiricism is based on three foundations: transparency, inspection and adaptation (Schwaber & Sutherland, 2013).

According to Schwaber and Sutherland (2013), transparency requires that the main aspects of the processes are visible and standardized so that interested parties have the same understanding, such as, the definition of a common language and what is meant as done. The inspection of the Scrum artifacts and the work progress must be done frequently to detect undesired deviations. Once detected, the process and product in development must be adjusted early to reduce future deviations.

#### 2.1.6.1 Scrum Team

The Scrum Teams are formed by the Product Owner, Development Team and the Scrum Master. They are self-organized, so they define a better way to work together without interference of external people, and are multifunctional, thus they have all skills to get the job done (Schwaber & Sutherland, 2013).

The Product Owner is the only responsible for the product backlog. Although there is a committee to evaluate the backlog, the Product Owner will be responsible for final decision making. The Development Team has the people responsible to deliver the final product. The Scrum Master is responsible to ensure that Scrum is understandable by the whole Scrum Team and being applied correctly, supporting the Product Owner, the Development Team and the management (Schwaber & Sutherland, 2013).

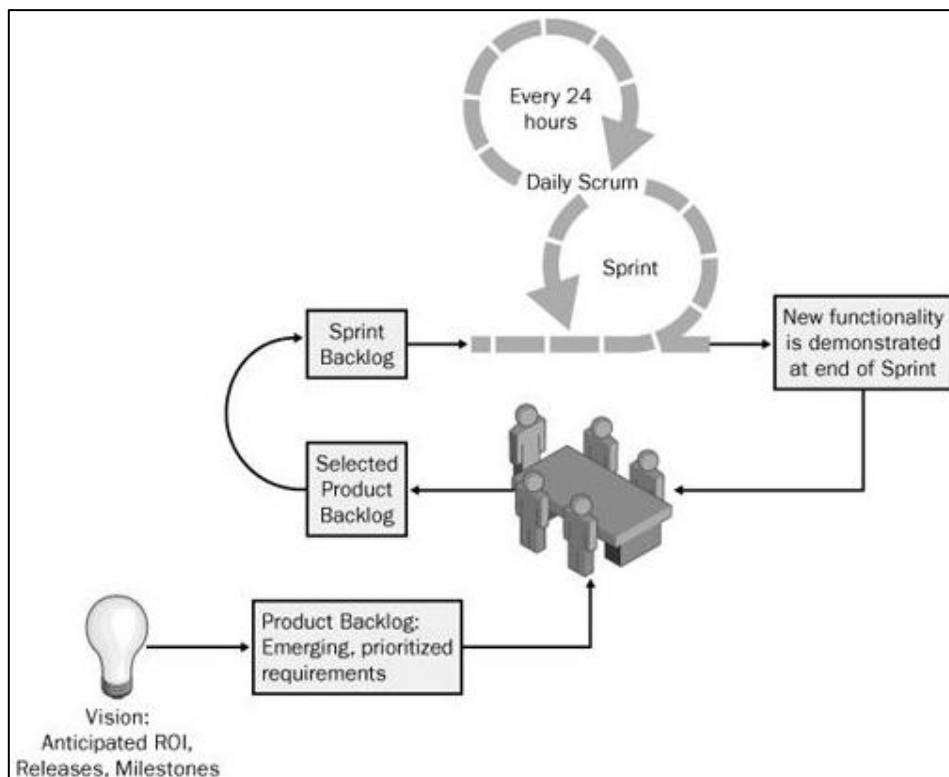
#### 2.1.6.2 Time-box Events and Sprint

Scrum uses time-box events to create regularity and avoid waste of time. Sprint is the heart of the Scrum. It is a one-month or least time-box within which an end product is delivered. The Sprint has five events: Sprint Planning Meeting, Daily Scrums, development work, Sprint Review Planning and Sprint Retrospective (Schwaber & Sutherland, 2013).

During the Sprint, any change that affects Sprint goal is made, the Development Team composition and the quality goals are kept constant and the scope must be clarified and renegotiated between the Product Owner and the Development Team as new information arise. Sprint can be cancelled, but it is not suggested (Schwaber & Sutherland, 2013).

### 2.1.6.3 Sprint Planning Meeting and Daily Scrum

In the Sprint Planning Meeting, the Product Owner presents the product backlog items prioritized according to the organizational goals. Then, the Development Team estimates which may be developed in the Sprint and how the work will be performed. The backlog items selected and the plan to develop them are called Sprint Backlog. The Daily Scrum is a daily meeting 15 minutes to inspect the progress of activities and create a plan for the next 24 hours (Schwaber & Sutherland, 2013). Figure 7 briefly illustrates an overview of framework Scrum from the definition of the vision to the delivery of a new functionality.



**Figure 7: Scrum Process Overview**

Source: Schwaber (2004).

#### 2.1.6.4 Sprint Review Meeting and Sprint Retrospective

In the Sprint Review Meeting, the Scrum Team and stakeholders come together to figure out what has or has not been delivered, the challenges faced, the lessons learned, and reset the product backlog, from the old and new items that arise. The Sprint Retrospective is a meeting held by the Scrum Team to inspect the last Sprint and propose adaptations to seek improvements in the next Sprint (Schwaber & Sutherland, 2013).

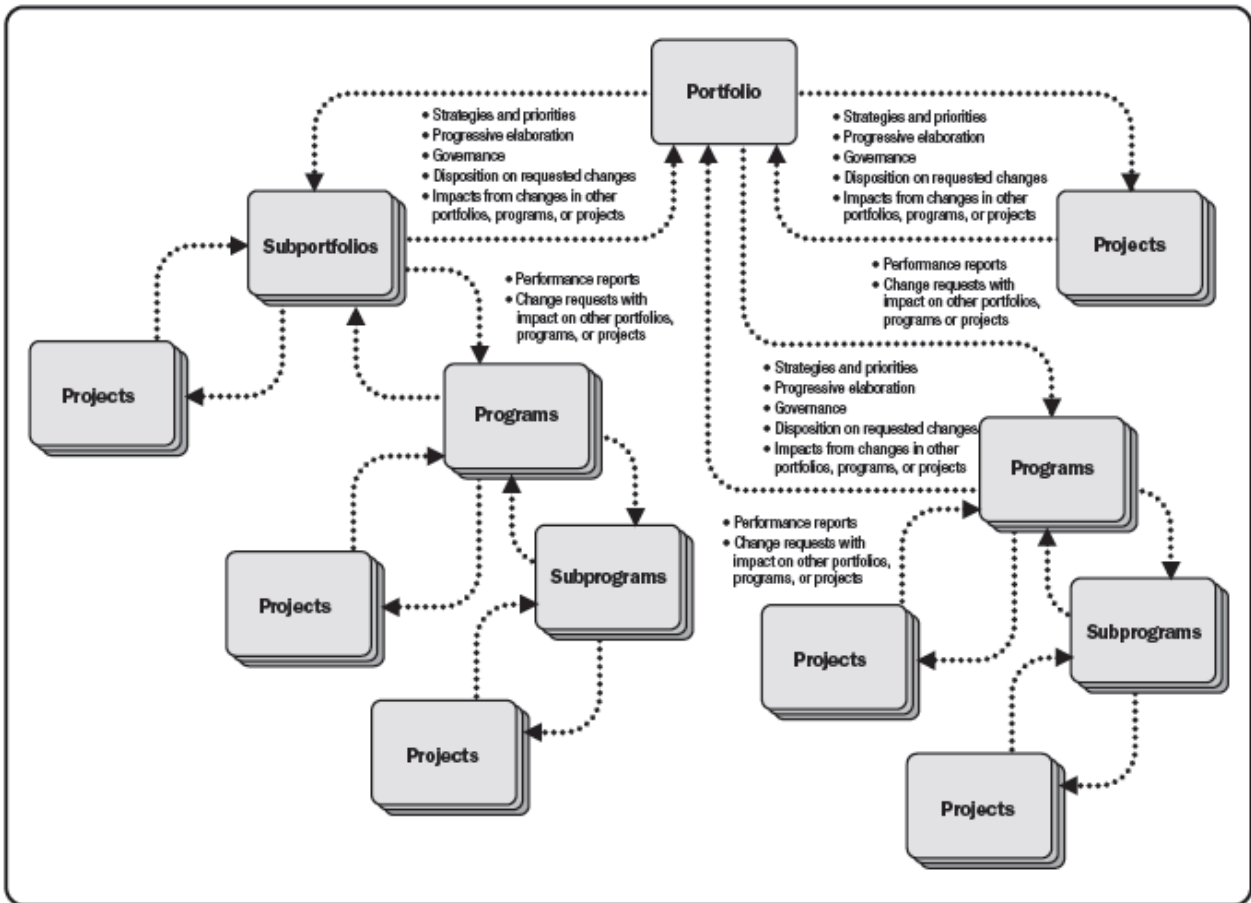
## 2.2 PROJECT PORTFOLIO MANAGEMENT

This chapter covers the main concepts of PPM, including, definitions, issues addressed, main objectives, key elements, impacts of adopting PPM, and PPM processes.

### 2.2.1 Project, Program and Portfolio Definition

For the Project Management Institute (PMI, 2013a, p. 3), “a project is a temporary endeavor undertaken to create a unique product, service, or result”. The key elements that define a project are: temporary, this is, the project has a defined beginning and end, and unique, i.e., the project has unique characteristics that distinguish it from other projects.

For PMI (2013c, p. 4), a program is defined as “a group of related projects, subprograms, and program activities that are managed in a coordinated way to obtain benefits not available from managing them individually”. PMI (2013b, p. 3) defines a portfolio as “a component collection of programs, projects, or operations managed as a group to achieve strategic objectives. The portfolio components may not be necessarily interdependent or have related objectives. The portfolio components are quantifiable, that is, they can be measured, ranked, and prioritized”. Figure 8 illustrates the interactions between portfolio, program, and project.



**Figure 8: Portfolio, Program, and Projects Interactions**

Source: PMI (2013).

### 2.2.2 Project, Program and Portfolio Management Definition

For PMI (2013a, p. 5), “project management is the application of knowledge, skills, tools, and techniques to meet the project requirements. Project management is accomplished through the appropriate application and integration of the 47 logically grouped project management processes, which are categorized into five Process Groups”. Kerzner (2009) defines project management “as the process of achieving project objectives through the traditional organizational structure and over the specialties of the individual concerned”.

For PMI (2013c, p. 6), “program management is the application of knowledge, skills, tools, and techniques to a program to meet the program requirements and to obtain benefits and control not available by managing projects individually. It involves aligning multiple components to achieve the program goals and allows for optimized and integrated cost, schedule, and effort”.



According to Cooper, Edgett and Kleinschmidt (1997a), the PPM is defined as a dynamic decision process in which projects of new products, and research and development (R&D) are constantly updated and reviewed. The projects are assessed, selected, and prioritized; current projects are advanced, canceled, or postponed; and resources are allocated and reallocated to the project activities. The PPM is all about the allocation of resources, the alignment between new projects and organizational strategy, and the balance between new projects, this is, dealing with risk versus return, maintenance versus growth, and short-term versus long-term.

For Reyck et al. (2005), the PPM considers the entire project portfolio the company is engaged with, in order to make decisions about which projects will be prioritized, added, or removed from the portfolio. For PMI (2013b, p. 5), the PPM “is the coordinated management of one or more portfolios to achieve organizational strategies and objectives. It includes interrelated organizational processes by which an organization evaluates, selects, prioritizes, and allocates its limited internal resource to best accomplish organizational strategies consistent with its vision, mission, and values”. Figure 9 illustrates the comparative overview of project, program, and portfolio management.

<b>Organizational Project Management</b>			
	<b>Projects</b>	<b>Programs</b>	<b>Portfolios</b>
<b>Scope</b>	Projects have defined objectives. Scope is progressively elaborated throughout the project life cycle.	Programs have a larger scope and provide more significant benefits.	Portfolios have an organizational scope that changes with the strategic objectives of the organization.
<b>Change</b>	Project managers expect change and implement processes to keep change managed and controlled.	Program managers expect change from both inside and outside the program and are prepared to manage it.	Portfolio managers continuously monitor changes in the broader internal and external environment.
<b>Planning</b>	Project managers progressively elaborate high-level information into detailed plans throughout the project life cycle.	Program managers develop the overall program plan and create high-level plans to guide detailed planning at the component level.	Portfolio managers create and maintain necessary processes and communication relative to the aggregate portfolio.
<b>Management</b>	Project managers manage the project team to meet the project objectives.	Program managers manage the program staff and the project managers; they provide vision and overall leadership.	Portfolio managers may manage or coordinate portfolio management staff, or program and project staff that may have reporting responsibilities into the aggregate portfolio.
<b>Success</b>	Success is measured by product and project quality, timeliness, budget compliance, and degree of customer satisfaction.	Success is measured by the degree to which the program satisfies the needs and benefits for which it was undertaken.	Success is measured in terms of the aggregate investment performance and benefit realization of the portfolio.
<b>Monitoring</b>	Project managers monitor and control the work of producing the products, services, or results that the project was undertaken to produce.	Program managers monitor the progress of program components to ensure the overall goals, schedules, budget, and benefits of the program will be met.	Portfolio managers monitor strategic changes and aggregate resource allocation, performance results, and risk of the portfolio.

**Figure 9: Comparative Overview of Projects, Program, and Portfolio Management**

Source: PMI (2013a).

Figure 9 illustrates the comparative overview of project, program, and portfolio management, considering the scope, change, planning, management, success, and monitoring.

### 2.2.3 Issues Addressed in Project Portfolio Management

The main issues faced by organizations in project selection and portfolio management are: disconnection between projects approved and strategic objectives; portfolios of low quality;

inefficiency in the decision process Go/Kill, leading to non-effective projects; scarce resources allocated to wrong projects; and oversimplification of product development (Cooper, Edgett & Kleinschmidt, 1997a).

Vähäniitty, Rautiainen, and Lassenius (2010) identified 34 research papers in the ScienceDirect portal related to inadequate portfolio management and/or the typical problems that occur in conjunction with inadequate portfolio management. They found eight problem areas that are symptomatic of inadequate portfolio management: i) excessive multitasking; ii) firefighting; iii) overload; iv) ineffective decision making, v) missing strategic alignment; vi) slipping schedules; vii) project failures and poor profitability; and viii) perceived need to improve project management.

They made a multiple-case study and found that all six small software organizations studied had problems in these areas, but just one, which had explicit portfolio management practices, had fewer problems than others which had not implemented any processes. They concluded that “explicit portfolio management seems relevant for small software organizations, at least when the development personnel possess multiple roles and responsibilities and are concurrently performing many different types of activities.”

#### 2.2.4 Value Maximization, Balance and Strategic Alignment

Cooper, Edgett and Kleinschmidt (1997a, 1997b) identified that the three main objectives of portfolio management in organizations are: value maximization, i.e., allocating resources to maximize the portfolio’s value in terms of some organizational goal; project balancing, and strategic alignment, i.e., all projects must be focused on organizational strategy.

Value maximization can be achieved through various methods, such as: expected commercial value, productivity index, dynamic rank ordered list and scoring models. The outcome of these methods is a prioritized list of projects that seek to achieve the expected goals of the organization (Cooper, Edgett & Kleinschmidt, 1997a, b).

For Cooper, Edgett and Kleinschmidt (1997a), balancing projects can be understood by analogy with balancing an investment portfolio. The graphical tools such as bubble charts and matrices (e.g.: Boston Consulting Group (BCG) Matrix adapted) are widely used to represent the balancing of a portfolio. In the research they identified that the companies surveyed have adopted

various forms of use of bubble diagrams, matrices, and other traditional charts (e.g.: pie chart) to properly balance their projects. There is no right or wrong, not even an unanimously considered an assertive and great way for balancing a project portfolio.

According to PMI (2013b), portfolio balancing supports the planning and allocation of resources (financial, human and physical assets) in accordance with the strategic direction and the ability to maximize the portfolio return based on the level of risk assumed by the organization. McFarlan (1981) considered three important elements that influence the materialization of risks: the size of the project, familiarity with the technology, and complexity of the project structure.

Cooper, Edgett and Kleinschmidt (1997a, b) affirmed that the strategy and resource allocation in new products should be closely connected. In practice, the strategy materializes when the effective allocation of resources occurs, thus consuming company's financial resources for the implementation of activities, this is, the strategy of the company is reflected where the money is spent. Scoring models, strategic buckets, and strategic checks can be used to link to business's strategy.

### 2.2.5 Key Elements of Project Portfolio Management

Cooper, Edgett and Kleinschmidt (1998) defined six key indicators to assess the degree of PPM in the companies studied, namely i) projects are aligned with the company's strategy, ii) portfolio contains projects with high added value, iii) expenses reflect the business strategy, iv) projects are finished on time, v) portfolio is balanced, and vi) portfolio has a proper number of projects.

The study results led the authors to ask themselves: what best performing companies were doing differently compared to companies with poor performance? The authors developed a scheme of portfolio performance indicators to answer this question. Two areas excel: portfolios balancing achieving the right balance of projects, and the right number of projects for the resources available.

In turn, Reyck et al. (2005) listed key elements of the revised PPM literature: centralized view of the project portfolio, financial analysis, risk analysis, interdependence among projects, resource constraints shared between projects (human resources, competence of staff, budget and

infrastructure); overall portfolio analysis, categorization, selection, accountability and governance, optimization, and specialized software PPM.

#### 2.2.6 Impacts Adopting Project Portfolio Management

Cooper, Edgett and Kleinschmidt (1998) found that senior managers in technology give more importance to PPM than marketing and sales managers, who in turn give more importance to the management of production and operation. The authors' conclusion is that companies that achieve a positive portfolio i) result in a balanced portfolio, strategically aligned with high added value, with the right number of projects and good response time, without delays, and ii) achieve a portfolio management process clearly defined and is supported by managers.

In their study, Reyck et al. (2005) concluded that the incremental adoption of PPM has a significant positive impact on the return of the project portfolio, and a significant negative impact on the number of problems reported in projects because of PPM was not adopted.

The study of Castro and Carvalho (2010b) concluded that the main aspects that differentiate organizations that perform the PPM of those who do not do so are: i) clarity about the availability of resources for management and implementation of projects, ii) evaluation, selection and prioritization of projects by category, iii) comparison and competition for resources projects for the same project category, iv) information from the ongoing projects is considered when assessing, selecting, prioritizing projects and allocating resources, and v) projects running are reassessed periodically, and may be paralyzed so that resources can be directed to other projects when necessary.

#### 2.2.7 Project Portfolio Management Processes

Cooper, Edgett and Kleinschmidt (1997b) suggested four decision processes: i) corporate planning – this is the process whereby the company's resources are allocated among business units (BUs), ii) strategy development at the BU level, iii) the BUs new product process – such as the use of a Stage-Gate® process, and iv) the portfolio review.

Castro and Carvalho (2010a, b) considered the following processes, which are most frequent in the literature: alignment with strategic priorities; definition of available resources;

classification of projects; projects evaluation; selection and prioritization of projects; allocation resources; and control of the portfolio. Petit and Hobbs (2010) defined as an assumption in their study that portfolio managers might also implement processes to manage and control uncertainties, besides monitor the changes as suggested in the literature.

PMI (2013b) mapped 16 portfolio management processes into three process groups (defining, aligning, and authorizing and controlling) and five knowledge areas (portfolio strategy management, portfolio governance management, portfolio performance management, portfolio communication management, and portfolio risk management). The process group should not be thought as a portfolio management phases. Figure 10 illustrates the three portfolio management process groups and the five knowledge areas, mapping 16 processes.

According to the PMI (2013b), the defining process group has processes performed to establish how the organizational strategy and objectives will be implemented in a portfolio; the aligning process group has processes performed to manage and optimize the portfolio; and the authoring and controlling process group has processes performed to establish how to authorize the portfolio and provide ongoing report.

Knowledge Areas	Process Groups		
	Defining Process Group	Aligning Process Group	Authorizing and Controlling Process Group
Portfolio Strategic Management	4.1 Develop Portfolio Strategic Plan 4.2 Develop Portfolio Charter 4.3 Define Portfolio Roadmap	4.4 Manage Strategic Change	
Portfolio Governance Management	5.1 Develop Portfolio Management Plan 5.2 Define Portfolio	5.3 Optimize Portfolio	5.4 Authorize Portfolio 5.5 Provide Portfolio Oversight
Portfolio Performance Management	6.1 Develop Portfolio Performance Management Plan	6.2 Manage Supply and Demand 6.3 Manage Portfolio Value	
Portfolio Communication Management	7.1 Develop Portfolio Communication Management Plan	7.2 Manage Portfolio Information	
Portfolio Risk Management	8.1 Develop Portfolio Risk Management Plan	8.2 Manage Portfolio Risks	

**Figure 10: Portfolio Management Process Groups and Knowledge Areas Mapping**

Source: PMI (2013b).

### 2.2.8 No Standard Approach for Project Portfolio Management

Cooper, Edgett and Kleinschmidt (1997b) understand that there have been advances in portfolio management, but much remains to be done. Finally, the allocation of resources, increasingly scarce, effectively, will make all the difference to achieve the objectives set by the organization.

According to them, there is no magic solution to the PPM, i.e., there is not an approach, model or standard method to solve the problems. In practice, they found that each company studied has been adopting and testing the means that best suit their needs. There was no evidence or interest in the use of mathematical programming, or optimization techniques. Financial methods were considered a problem to prioritize projects, since financial data is highly inaccurate, especially in the beginning of the projects. The adoption of tools and decision support systems (DSS) was the way used by the organizations studied to support managers in their decision making.

Archer and Ghasemzadeh (1999) proposed an integrated framework for project portfolio selection to support decision makers. They argued that support tools are essential to implement each technique used, and the decision makers are free to select specific techniques. They suggested that the set of main stages in the framework can be integrated into a DSS, composed essentially of i) a project portfolio database management; ii) a model management module to support the techniques or models to be used; and iii) a user interface to the model management and database management modules. Ghasemzadeh and Acher (2000) developed a prototype of a DSS and found that users perceived it as a useful tool for project portfolio selection and easy to use, but additional research is necessary to extend their work.

In accordance to Reyck et al. (2005), organizations do not need to take all the key elements of the PPM to generate benefits. They suggest that a selection of the elements according to each context and level of maturity can be done for the adoption of PPM. Machado et al. (2011) identified that the effective implementation of PPM still leaves some gaps, less than half of the respondents indicated the existence of PPM processes deployed in their companies, and, for the most part, the projects are still centralized and prioritized by high management without a clear definition of an area responsible for collecting, analyzing and distributing the projects' information.

### 2.2.9 Portfolio Management and Agile Software Development

Rautiainen, Shantz and Vähäniitty (2011) developed a descriptive case study to show how the company has introduced project portfolio management to help them scale agile software development. The case had suffered with long time-to-market due to thrashing, which was caused by frequently changing priorities due to an ad-hoc prioritization process and handovers.

The findings showed that the company introduced a model for managing the portfolio of projects in a structured way that allowed the company to reduce the number of ongoing projects, reducing thrashing, providing visibility of the ongoing activities, and helping coordinate the work of multiple Scrum teams. Scaling software agility was also studied by Leffingwell (2007).

Santos Filho (2012) identified 14 interactions points between the PPM processes and Scrum principles and practices, based on the literature review, and a case study in the public sector. Table 4 illustrates these points. The findings suggested that there are no problems related to the use of Scrum due to the adoption of PPM processes, and, additionally, were identified facilitating practices to the interactions between processes. This result is quite similar with the findings of Rautiainen, Shantz and Vähäniitty (2011), which presupposes that Scrum and PPM can coexist together.

**Table 4: Interfaces between PPM and Scrum**

Resources required by a new component.	Running component to identify components.
Deadlines for a new component.	Performance of ongoing component for review and communication.
Risks related to a new component.	Progress reports of ongoing component for monitoring and controlling risks.
New component for managing scope.	Data of ongoing components to support the process of risk management.
New component to prevent interruptions.	Human resource capacity and production for component selection.
Initiation planning component.	Human resource capacity and production for balancing.
Initiation and deactivation of components for (re)allocation of resources.	Capacity and resource allocation for review and reporting of portfolio.

Source: adapted from Santos Filho (2012).



## 2.3 UNCERTAINTY AND DYNAMIC CAPABILITIES

This section describes the concepts of uncertainty and dynamic capabilities, the conceptual framework proposed by Petit and Hobbs (2010), and the main types of uncertainties found in their research.

### 2.3.1 Uncertainty

According to Petit and Hobbs (2010, p. 46), the PPM literature “does not provide an adequate framework for the study of changes to the project portfolio between periodic review cycles, a phenomenon that is common in dynamic environments”. They reviewed the literature on management of uncertainty, and proposed a novel framework based on concepts derived from the dynamic capabilities and sensemaking. The assumption for the research is that portfolio managers might also implement processes to manage and control uncertainties besides monitor the changes.

Petit and Hobbs (2010) explained the differences between risks, changes, deviations, and unexpected events according to the literature. Table 5 illustrates these differences. They found that some authors have advocated using the concept of uncertainty management instead of risk management, because it is not just about managing threats and opportunities, but also exploring and understanding the origins of project uncertainty before focusing on its management.

**Table 5: Differences between Risks, Changes, Deviations, and Unexpected Events**

Element	Definition	Author
Risk	An uncertain event or condition that, if it occurs, has a positive or negative effect on one or more project objectives such as scope, schedule, cost, and quality.	PMI
Changes	Realized situations with a significant divergence to the project plan.	Hällgren and Maaninen-Olsson
Deviations	A situation, regardless of consequence—positive or negative, large or small—that deviates from any plan in the project.	Hällgren and Maaninen-Olsson
Unexpected events	Reopening caused by stakeholders redefining some of the project parameters, revisions to plan to improve its accuracy and adapt to events, and finally daily fine-tuning.	Söderholm

Source: adapted from Petit and Hobbs (2010).

### 2.3.2 Dynamic Capabilities

Eisenhardt and Martin (2000, p. 1118) concluded that dynamics capabilities are “well-known organization and strategic process like alliancing and product development whose strategic value lies in their ability to manipulate resources into value-creating strategies... Their broad structural patterns vary with market dynamism, ranging from the robust, grooved routines in moderately dynamics markets to fragile semi-structured ones in high-velocity ones. They evolve via well-known learning mechanisms”.

For Teece (2007, p. 1319), “dynamic capabilities can be disaggregated into the capacity (1) to sense and shape opportunities and threats, (2) to seize opportunities, and (3) to maintain competitiveness through enhancing, combining, protecting, and, when necessary, reconfiguring the business enterprise’s intangible and tangible assets. Dynamic capabilities include difficult-to-replicate enterprise capabilities required to adapt to changing customer and technological opportunities”.

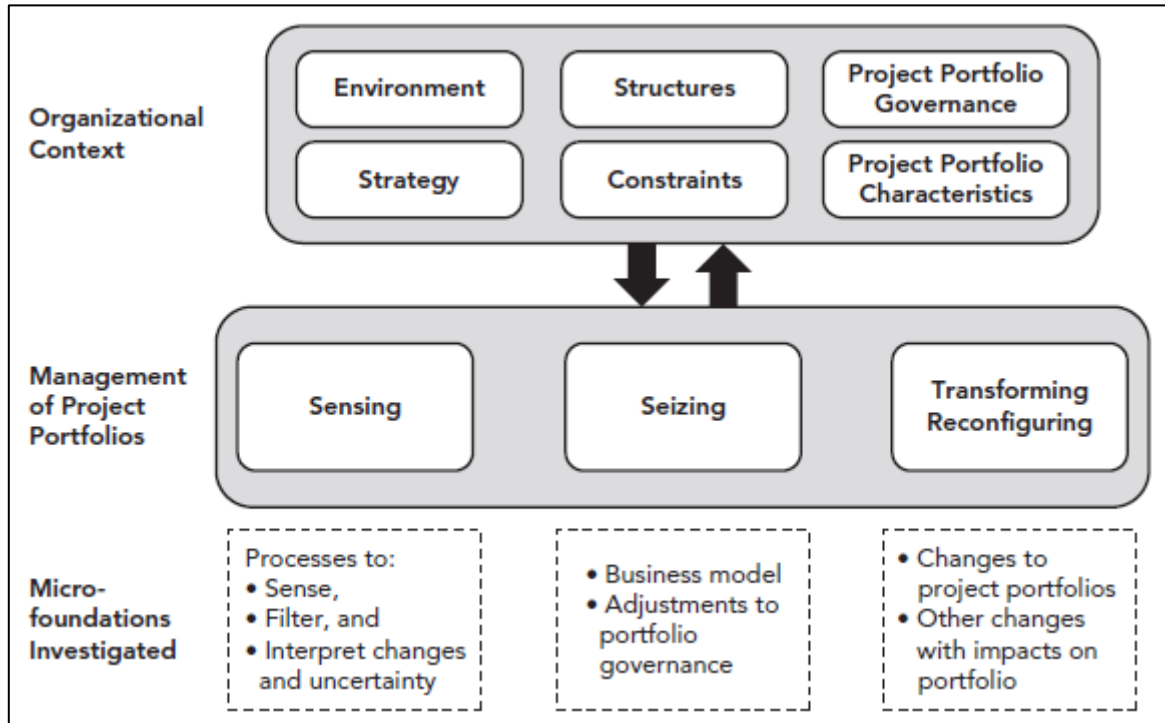
Petit and Hobbs (2010) found in some studies of dynamic capabilities that to achieve a strategic advantage it is necessary to reallocate and re-optimize resources and capabilities to adapt to a changing environment, instead of just developing unique resources or capabilities, as proposed in the resource-based view.

### 2.3.3 Conceptual Framework

Petit and Hobbs (2010) created a conceptual framework based on Teece’s dynamic capabilities framework and Weick’s sensemaking to study the management of uncertainty in project portfolios. It is formed by three components: i) organizational context, 2) dynamic capabilities, and iii) the micro-foundations to be investigated, which are depicted in Figure 11.

The organizational context is analyzed to understand why project portfolio is running and under which organizational constraints. Dynamic capabilities are divided into three groups of capabilities: i) sensing: the organization information processing mechanism to identify trends, events, competitors, markets, changing customer needs, technologies, etc.; ii) seizing: the structures, procedures, and rules to decide what to do in face of changes and uncertainty; and iii)

transforming and reconfiguring: the organization might have to change the organizational routines, adapting, to face a changing environment.



**Figure 11: The Conceptual Framework to Study the Management of Uncertainty in Project Portfolios**

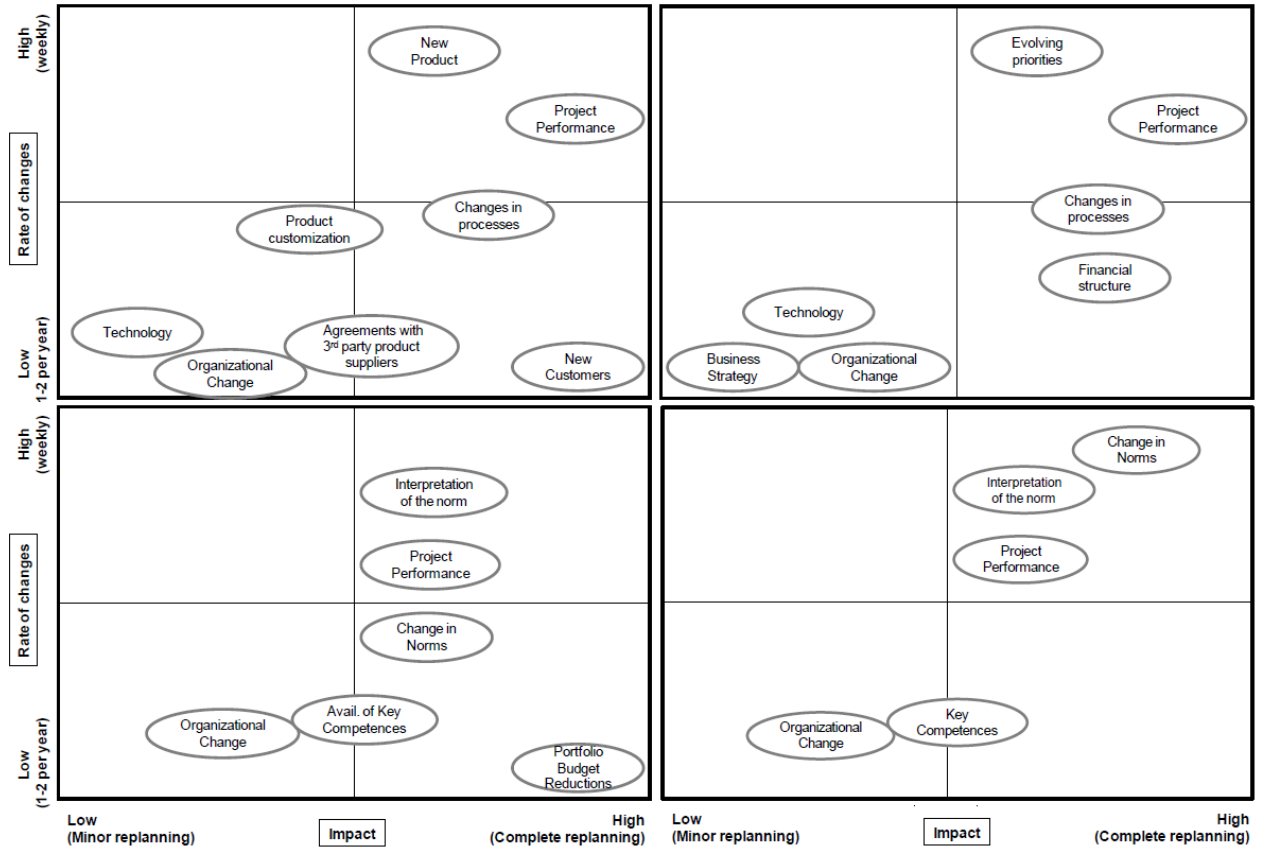
Source: adapted from Petit and Hobbs (2010).

#### 2.3.4 Types of Uncertainty

Petit and Hobbs (2010) and Petit (2011) identified many additional types and sources of change the organizations managing the project portfolio were facing. They also identified the rate of change and impact of these new types of changes. According to them, “the main source of uncertainty is related to scope changes, which are all constantly evolving in turbulent market”.

The types of changes identified in their study were: new product (scope), project performance, changes in processes, need for customization, new customers and new market (scope), changes in agreements with third-party suppliers (scope), structural re-organizations/organizational change, technology, evolving priorities (scope), financial structure, changes in business strategy, interpretation of the norm (scope), changes in norm, portfolio budget reduction, and key competences.

Figure 12 illustrates four charts, one for each portfolio studied, according to the rate of change and impact. Almost all scope changes are of high impact and high rate of change.



**Figure 12: Types of Changes, Rate of Change, and Impact on the Four Portfolios Studied**

Source: adapted from Petit (2011).

### 2.3.5 Project Complexity and Uncertainty

There are a variety of ways to evaluate the complexity and uncertainty of a project. Carvalho and Rabechini (2011) exemplify at least five models created by Crawford, Lewis, Sabbag, Maximiano, and Shenhar and Dvir. Little (2005) developed a system to score project complexity based on six attributes: team size, mission criticality, team location, team maturity, domain knowledge gaps, and dependencies. Table 6 illustrates all six attributes to be evaluated according to the definition that most closely match the project context. The scale goes from 1 to

64 and a score above 15 is high. The project's overall complexity is calculated based on the individual aggregation, according to the formula:

$$2^{\sum \log_{10} x_i}$$

, where  $i$  is the complexity attribute and  $x_i$  is the individual complexity score. In effect, the log  $x$  terms are scaled information measures.

For example, an IT project has 50 people working in the team in the same building; 1000 potential users of the product to be delivered; developers know the domain partially; and the resources are shared with others projects. It is possible to set, for each attribute defined in Table 6, the following values: Team size = 10, Mission criticality = 7, Team location = 3, Team capacity = 5, Domain knowledge gaps = 5, and Dependencies = 7. Applying the formula above, it is possible to achieve the result of 24.

**Table 6: System to Score Project Complexity**

Attribute	Complexity Score (from 1 = minimally complex to 10 = highly complex)				
	1	3	5	7	10
Mission criticality	Speculative	Small user base	Established market	Mission-critical with large user base	Safety-critical with significant exposure
Team location	Same room	Same building	Within driving distance	Same time zone +/- 2 hrs.	Multisite, worldwide
Team capacity	Established team of experts	New team of experts	Mixed team of experts and novices	Team with limited experience and a few experts	New team of mostly novices
Domain knowledge gaps	Developers know the domain as well as expert users	Developers know the domain fairly well	Developers require some domain assistance	Developers have exposure to the domain	Developers have no idea about the domain
Dependencies	None	Limited, well insulated	Moderate	Significant	Tight integration with several projects
Team size	1	5	15	40	100

Source: adapted from Little (2005).

Little (2005) also developed a system to score project uncertainty based on four attributes: market uncertainty, technical uncertainty, project duration, and other project's dependencies on that project and scope flexibility. The scale goes from 1 to 16 and a score above 6 is high. The

project's overall uncertainty is calculated using the same formula applied to calculate the project's overall complexity, but with the uncertainty attributes. Table 7 illustrates this system and presents all four attributes to be evaluated according to the definition that most closely matches the project context.

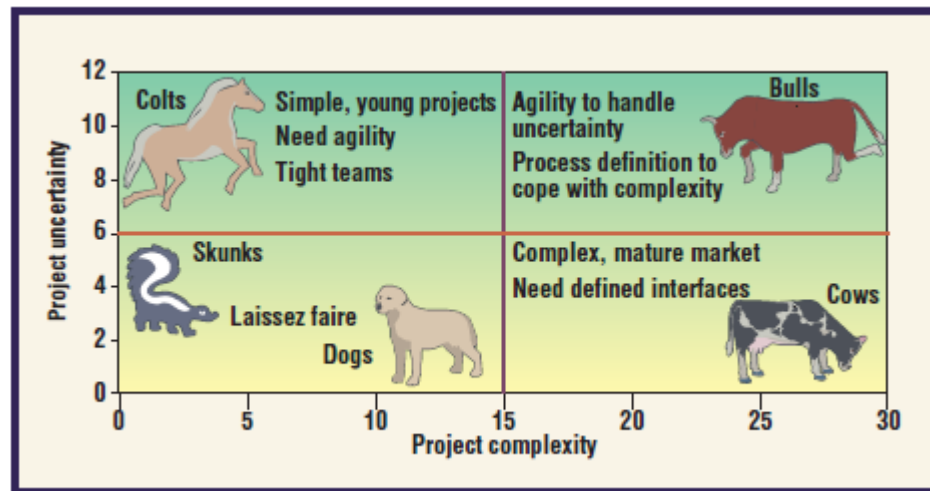
**Table 7: System to Score Project Uncertainty**

Attribute	Uncertainty Score (from 1 = minimally uncertain to 10 = highly uncertain)				
	1	3	5	7	10
Market uncertainty	Known deliverable, possibly defined contractual obligation	Minor changes in market target expected	Initial guess of market target likely to require steering	Significant market uncertainty	New, unknown, and untested market
Technical uncertainty	Enhancements to existing architecture	We think we know how to build it	We're not quite sure if we know how to build it	Some incremental research involved	New technology, new architecture; might be some exploratory research
Project duration	1-4weeks	6 months	12 months	18 months	24 months
Dependencies, scope flexibility	Well-defined contractual obligations or infrastructure with published interfaces	Several interfaces Scope isn't very flexible	Scope has some flexibility	Some published interfaces Scope is highly flexible	No published interfaces

Source: adapted from Little (2005).

For example, the same IT project has significant market uncertainty; the developers think they know how to build the solution; needs 20 months to be completed; and the scope is partially known and there many changes. It is possible to set, for each attribute defined in Table 7, the following values: Market uncertainty = 7, Technical uncertainty = 3, Project duration = 7, and Dependencies, scope flexibility = 7. Applying the formula above, it is possible to achieve the result of 12.

Little (2005) borrowed the concepts of the BCG to create his own matrix, called Houston Matrix. Using the results calculated previously, it is possible to frame the end result in the matrix, as illustrated in Figure 13.



**Figure 13: Houston Matrix Quadrant Assessment**

Source: Little (2005).

Figure 13 illustrates the four quadrants of Houston matrix: Dogs are simple projects with low uncertainty, such as, mature projects being developed by small teams; Colts are simple projects with high uncertainty, such as new products with market and technical uncertainty; Cows are complex projects with low uncertainty, such as mature projects being developed by large teams; and Bulls are complex projects with high uncertainty, such as next-generation products.

According to the results of the example, i.e., project complexity = 24 and project uncertainty = 12, it is possible to conclude that this project can use some agile methodology to handle uncertainties, according to the Houston Matrix proposed by Little (2005).

## 2.4 SUMMARISING

This section presented the literature review about the three theoretical poles that support this research. Agile practices grew out of a need to manage projects characterized by complexity and uncertainty in a turbulent, speedy, risky, and unpredictable environment. Section 2.1 described the main principles and practices of the known agile methodologies. Section 2.2 showed the concepts of PPM, and section 2.3 briefly the concepts of uncertainty and dynamic capabilities and a framework to study the management of uncertainty in project portfolios.

### 3 METHODOLOGY

For Martins and Theóphilo (2009), the scientific knowledge results from systematic and methodical investigation of reality. It analyses facts and phenomena to discover their causes and concludes on general laws and is bounded by the need of concrete evidence. The sciences are classified in two groups: formals and factual, according to their content. Except for Logic and Mathematic, all other fields of knowledge are classified in factual sciences, also called experimental or empirical. They study concrete objects and depend on experimental tests of their hypothesis. The factual sciences are divided in natural and social sciences. Gil (2008, p. 26) defines social research “as a process that, using scientific methodology, allows to obtain new knowledge in the field of social reality”.

The literature defines that knowledge generation is based on four poles: epistemological, theoretical, methodological, and technical, in that order. The epistemological polo drives the theoretical polo, which in turn drives the methodological polo that influences the technical polo (Martins & Theóphilo, 2009). All these poles are mentioned in this section.

The epistemological polo exercises a critical supervision role on research. It considers the research problem, the production of the scientific object, validity and reliability of the measurement instruments (Martins & Theóphilo, 2009).

According to Martins and Theóphilo (2009), the research problem allows submitting reality to systematic investigation. It arises from restlessness, doubt, and curiosity about an unanswered question. The research begins with the problem and its logic is oriented by the search for a solution to the problem. To formulate an adequate scientific problem, the following conditions are sufficient and necessary:

- I. It must be accessible to the scientific field in which the problem can be inserted in such a way that it can be treated.
- II. It has to be well defined, in the sense that it tends to a unique solution and, having all relevant explicit elements, it rises investigations that can be carried out to solve it.
- III. Its structure and, particularly, its assumptions are not false.



- IV. It has to formulate conditions in advance about the type of solution and type of evidence that would be acceptable (Martins & Theóphilo, 2009).

The research problem of this study is the link between agile software development and project portfolio management in dynamic environments. As showed before, there are few studies on these subjects.

### 3.1 RESEARCH STRATEGY

This section describes the research strategy classification and type, and an overview of the research process.

#### 3.1.1 Research Strategy Classification

According to Martins and Theóphilo (2009, p. 37), “the aim of the methodology is the improvement of the procedures and criteria used in the research.” The methodology addresses the way science tries to capture reality. The scientific method provides research strategies with general and specific techniques that allow producing work according to scientific standards and of true scientific value.

For Gil (2008), the classification of social research into three groups proposed by Selltitz is the most widely adopted nowadays, he also used it with a small change in nomenclature, which is the same used by Yin (2009). The groups are: exploratory research, descriptive research, and explanatory research.

“The exploratory research mainly aim to develop, clarify and change concepts and ideas, in view of the formulation of accurate problems or searchable hypotheses for further studies... the descriptive research has as primary objective the description of the characteristics of a given population or phenomenon, or establishing relationships between variables... the explanatory research has a central concern to identify the factors that determine or contribute to the occurrence of phenomena.” Gil (2008, p. 27).

The exploratory research is done especially when the theme chosen is underexplored and it becomes difficult to formulate precise and practicable hypotheses. It usually involves bibliographical and documentary research, not standardized interviews, and case study (Gil, 2008).

The theme chosen for this research is the agile software development relating to PPM in dynamic environments and, as shown in chapter 1 and 2, there are few studies about it and great opportunities for further exploration. Since there are few studies linking agile software development and PPM, this investigation has the characteristics of an exploratory research.

### 3.1.2 Type of Research Strategy

To define the appropriate type of strategy for a study, Yin (2009) suggests that three conditions be analyzed:

1. The type of research question posed;
2. The extent of control an investigator has over actual behavioral events; and
3. The degree of focus on contemporary as opposed to historical events (Yin, 2009, p. 8)

Table 8 correlates five known strategies in social sciences with each condition to help the researcher define the appropriate research strategy for the study.

**Table 8: Relevant Situations for Different Research Strategies**

<b>Method</b>	<b>(1) Form of Research Question</b>	<b>(2) Requires Control of Behavioral Events?</b>	<b>(3) Focuses on Contemporary Events?</b>
Experiment	How, why?	yes	yes
Survey	Who, what, where, how many, how much?	no	yes
Archival Analysis	Who, what, where, how many, how much?	no	yes/no
History	How, why?	no	no
Case Study	How, why?	no	yes

Source: adapted from Yin (2009).

Defining the research question is the most important step in scientific research. It is necessary to be patient and take the time to develop it (Martins and Theóphilo, 2009; Yin, 2009). The research question of this study is formulated as *How agile software development contributes to project portfolio management in dynamic environments of an integrated solutions provider?* It is a *How* question; the study addresses events and variables over which the investigator has no control; and there are few studies on the theme that can be examined, which focus on contemporary events.

For Yin (2009, p. 13), case study is appropriate when “a ‘how’ and ‘why’ question is being asked about a contemporary set of events, over which the investigator has little or no control”. Consequently, according to the conditions assessed above, case study appears to be an adequate strategy to answer the research question. Other strategies being applied to this study include bibliographical and documentary research.

#### 3.1.2.1 Case Study

Martins and Theóphilo (2009) define a case study research as an empirical investigation to study phenomena within their real context, where the researcher has little or no control over events and variables, seeks to understand the whole context of the case, describing and interpreting it. Yin’s (2009) definition of a case study is based on its scope and other technical characteristics, such as data collection and data analysis:

1. A case study is an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.
2. The case study inquiry copes with the technically distinctive situation in which there will be many more variables of interest than data points, and as one result relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result benefits from the prior development of theoretical propositions to guide data collection and analysis. (Yin, 2009, p. 18)

According to Yin (2009, p. 66), “case study research is among the hardest types of research to do because of the absence of routine procedures”. He suggests five types of case study preparations to perform a good research: define the investigator’s desired skills, provide training for a specific case study, develop a case study protocol, the screening of candidate cases, and conduct a pilot case study. He lists these commonly required skills:

- A good case study investigator should be able to ask good questions-and interpret the answers.
- An investigator should be a good "listener" and not be trapped by her or his own ideologies or preconceptions.
- An investigator should be adaptive and flexible, so that newly encountered situations can be seen as opportunities, not threats.
- An investigator must have a firm grasp of the issues being studied, even if in an exploratory mode. Such a grasp reduces the relevant events and information to be sought to manageable proportions.
- A person should be unbiased by preconceived notions, including those derived from theory. Thus, a person should be sensitive and responsive to contradictory evidence. (Yin, 2009, p. 69).

The section 3.3 describes the case selection and the section 3.4 the development of a case study protocol. Considering the time constraint of this research, a pilot case study was not considered.

#### 3.1.2.2 Bibliographical Research

Martins and Theóphilo (2009) define a bibliographical research as a strategy needed for any scientific research. The goal is to understand, analyze, and discuss a subject or problem from a theoretical framework. The researcher must start by searching for reference books (usually classics) and after that look for articles in magazines and journals, to find recent studies. Chapter two presents a literature review of research from books, guides, thesis, dissertations, and articles in magazines and journals.

### 3.1.2.3 Documentary Research

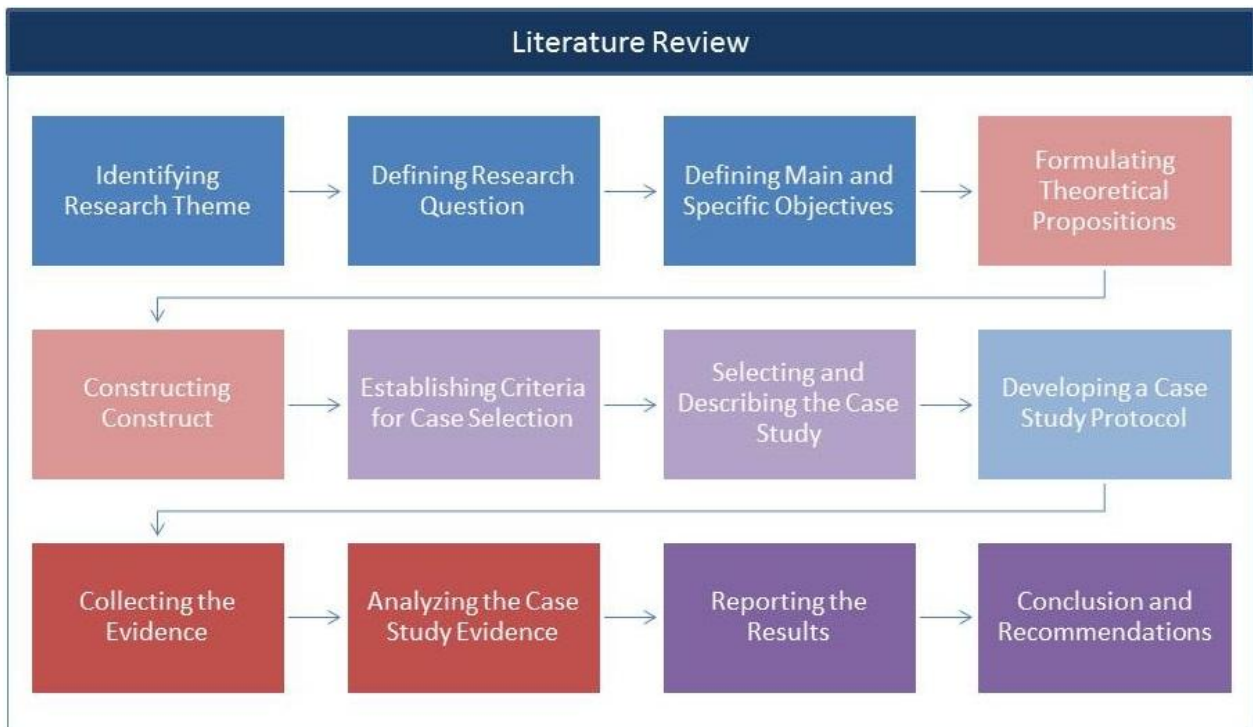
Documentary research is a research strategy that uses any type of documents as data source, information and evidences (Martins & Theóphilo, 2009). With this approach request for proposals (RFP), proposals, contracts, plans, schedules, technical and functional specifications, reports, e-mails, and others project documents were analyzed.

Summarizing, the research strategy for this study has the following characteristics:

- Classified as a qualitative research, where facts and phenomena are understood, interpreted, and described;
- Exploratory;
- Conducted over a single-case study, as defined in the section 3.3.1; and
- Based on bibliographical and documentary research.

### 3.1.3 Overview of the Research Process

On Figure 14 an overview of the research process based on Martins & Theóphilo (2009) and Yin (2009) is shown. The following sections describe in more detail each phase of the process, except for the identification of the research theme, defining the research question and the main and specific objectives, which have been previously outlined in this study.



**Figure 14: Overview of the Research Process**

Source: created by the author.

Figure 14 illustrates all phases of the research process, which are constantly over influence of the literature review.

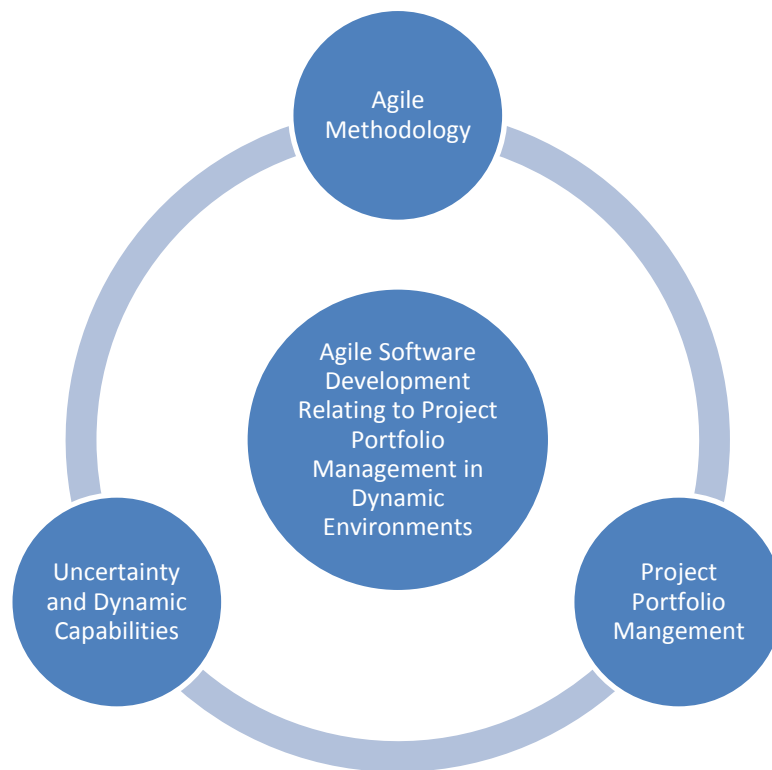
### 3.2 FORMULATING THEORETICAL PROPOSITIONS AND CONSTRUCTING CONSTRUCT

The theoretical polo orients the definitions of hypothesis and the construction of constructs. It is the place of models, theories, hypothesis, theses, concepts, definitions, and constructs (Martins & Theóphilo, 2009). For Kerlinger (as cited in Martins & Theóphilo, 2009, p. 28), “a theory is a set of constructs (concepts), definitions, and propositions related to each other, that present a systemic view of phenomena by specifying relationships among variables, with the purpose of explaining and predicting phenomena of reality”.

### 3.2.1 Formulating Propositions

A hypothesis is a proposition with a sense of assumption that anticipates an answer to a problem that can be accepted or rejected through the theory and the results of the analysis of quantitative and qualitative information, data, and evidence. The research problem, research question, objectives, and the hypothesis are the essence of a scientific research (Martins & Theóphilo, 2009). For Yin (2009, p. 28), “each proposition directs attention to something that should be examined within the scope of study”. The proposition refers to an important theoretical issue and suggests the investigator where to look for relevant evidence.

The pillars defined from the literature review are illustrated in Figure 15 and give theoretical support to the agile software development relating to project portfolio management in dynamic environments.



**Figure 15: Pillars Defined from the Literature Review**

Source: created by the author.

The pillars identified in Figure 15 were used to identify the main references in the literature and the relevant concepts that supported the formulation of propositions and the construction of the construct. These are described in the following sections.

#### 3.2.1.1 Agile Methodologies

These are the propositions formulated based on the literature review of agile methodology:

- **Proposition 1:** Processes and tools are important, but talented, motivated, valued, skilled, self-organized people and the interaction between them are far more important, thus process and tools must be adapted to people (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006).
- **Proposition 2:** Working software delivered early and frequently allows for continued wins, early feedback from users, rapid response to changing marketing conditions, so that the customer can sense that deliverables are evolving, despite of what was documented (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006, Leffingwell, 2007).
- **Proposition 3:** Just the customer collaboration and close communication with the development team during the time of delivery can provide the customers with their real needs and prevent undesirable outcomes (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006).
- **Proposition 4:** Planning is useful, important and necessary, but adapting to changes to the plan is more important and useful for the customer's competitive advantage, especially in an environment characterized by change, speed, and turbulence (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006, Leffingwell, 2007).

#### 3.2.1.2 Project Portfolio Management

These are the propositions formulated based on the literature review of PPM:

- **Proposition 5:** A portfolio is a collection of programs, projects, or operations managed as a group to achieve organizational strategies and objectives (PMI, 2013).



- **Proposition 6:** The project portfolio management is defined as a dynamic decision process where projects are evaluated, selected, prioritized, authorized, assessed, and monitored (Cooper, Edgett & Kleinschmidt, 1997a; Reyck et al., 2005; PMI, 2013b).
- **Proposition 7:** The effective allocation of resources, increasingly scarce, will make all the difference in achieving the objectives set by the organization (Cooper, Edgett & Kleinschmidt, 1997a; Reyck et al., 2005; PMI, 2013b).

### 3.2.1.3 Uncertainty and Dynamic Capabilities

These are the propositions formulated based on the literature review of uncertainty and dynamic capabilities:

- **Proposition 8:** Portfolio management might implement process to manage and control uncertainty, and not only monitor changes (Petit & Hobbs, 2010; Petit, 2011).
- **Proposition 9:** There are additional types and sources of change that the organizations managing the project portfolio were facing, beyond the commonly described in the literature (Petit & Hobbs, 2010).
- **Proposition 10:** It is necessary to reallocate and re-optimize resources and capabilities to adapt to changing environments (Petit & Hobbs, 2010; Petit, 2011).

### 3.2.2 Constructing the Construct

For Martins and Theóphilo (2009, p. 35), “a construct is a variable – set of terms, concepts, and variables, that is, a robust operational definition that seeks to empirically represent a concept within a specific theoretical framework”. The researcher should identify the observable and measurable variables that can be represented by theoretical variables.

Table 9 shows the constructs created to link the research question, the propositions defined in the section before, and the evidences to be collected, supporting the analysis, conclusion, and contributions.

**Table 9: Research Construct**

<b>Research Question:</b> <i>How agile software development contributes to project portfolio management in dynamic environments of an integrated solutions provider?</i>				
<b>Research Objectives</b>	<b>Pillars</b>	<b>Propositions</b>	<b>Variables</b>	<b>Questions</b>
to identify the common practices in the field of agile software development and project portfolio management adopted by the company;	Agile Methodology	Processes and tools are important, but talented, motivated, valued, skilled, self-organized people and the interaction between them are far more important, thus process and tools must be adapted to people (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006).	team organization, team expertise, organizational climate, knowledge gaps	1, 2, 3, 4
		Working software delivered early and frequently allows for continued wins, early feedback from users, rapid response to changing marketing conditions, and customer can sense that deliverables are evolving, despite of what was documented (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006, Leffingwell, 2007).	frequency of releases, processes of evaluation and integration, project's duration	5, 6, 7, 8
		Just the customer collaboration and close communication with the development team during the time of delivery can provide the customers with their real needs and prevent undesirable deliverables (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006).	people's interactions, communication of changes, market needs, definition of scope	9, 10, 11, 12
		Planning is useful, important and necessary, but adapting to changes to the plan is more important and useful for the customer's competitive advantage, especially in an environment characterized by change, speed, and turbulence (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006, Leffingwell, 2007).	development plan, review of plan, ways of facing changes	13, 14, 15
to identify the types of uncertainties found in the portfolio studied;	Project Portfolio Management	A portfolio is a collection of programs, projects, or operations managed as a group to achieve organizational strategies and objectives (PMI, 2013).	portfolio and projects description, project's interdependences, criticality to business	16, 17, 18, 19
		The project portfolio management is defined as a dynamic decision process where projects are evaluated, selected, prioritized, authorized, assessed, and monitored (Cooper, Edgett & Kleinschmidt, 1997a; Reyck et al., 2005; PMI, 2013b).	processes of PPM, decision makers, criteria of prioritization	20, 21, 22
		The effective allocation of resources, increasingly scarce, will make all the difference to achieve the objectives set by the organization (Cooper, Edgett & Kleinschmidt, 1997a; Reyck et al., 2005; PMI, 2013b).	criteria, planning, communication for HR allocation, team size, geographical distribution	23, 24, 25, 26, 27
to make recommendations according to the results of the research and literature review.	Uncertainty and Dynamic Capabilities	Portfolio management might implement process to manage and control uncertainty, and not only monitor changes (Petit & Hobbs, 2010; Petit, 2011).	processes to manage uncertainty, process to handle changes	28, 29
		There are additional types and sources of change that the organizations managing the project portfolio were facing, beyond the commonly described in the literature (Petit & Hobbs, 2010).	sources of uncertainty, types and frequency of changes	30, 31, 32
		It is necessary to reallocate and re-optimize resources and capabilities to adapt to changing environments (Petit & Hobbs, 2010; Petit, 2011).	development of HR and dynamic capabilities, training of HR, domain technological	33, 34, 35

Source: created by the author.

### 3.3 CASE SELECTION

This section presents the single-case design, establishing criteria for case selection, and the case selected.

#### 3.3.1 Single-Case Design

Yin (2009) presents four types of case study designs: single-case (holistic) design, single-case (embedded) design, multiple-case (holistic) design, and multiple-case (embedded) design. He appoints five rationales to justify the adoption of single-case:

- It represents the critical case in testing a well-formulated theory, which has a clear set of propositions;
- It represents an extreme or unique case;
- It is a typical case;
- It is a revelatory case: the researcher has a chance to observe and analyze a phenomenon previously inaccessible to social science inquiry; and
- It is a longitudinal case: studying the same single case at two or more different points in time.

This research is conducted over a single-case (holistic) design – one project portfolio – that is adherent to almost rationales described above. The company studied is a unique case with specific particularities considering its market, it is a typical case IS provider. The researcher had the opportunity to observe and analyze the organizational context and the projects portfolio in depth, that would not have been possible otherwise. Furthermore, this study has a time constraint related to the completion of the master research.

### 3.3.2 Establishing Criteria for Case Selection

The following criteria were defined to select the unit to be analyzed, partially based on criteria used by Petit (2011) in his research, and considering some objectives of this study:

- Organization should have dynamic environments with a high level of uncertainty and/or high volume of changes to their project portfolio;
- Organization must be applying any agile methodology approach, preferably, Scrum;
- There is easy access to key people and documents;
- Projects of the portfolio must be long-term (at least one year);
- Portfolio must be developed for external clients (preferably with fixed-price contracts); and
- Portfolio must have projects sharing, partially or totally, the same human resources.

### 3.3.3 Case Selected

One portfolio in one organization was selected for this research. The case is described in more detail in section 4.1. The organization, nicknamed *Company Brazil*, is a subsidiary of a multinational company dedicated to the research, development, and integration of technological solutions, including software and hardware. It has three BUs, all focused on providing custom services.

One of the BUs offers integrated solutions that include that include customization and integration of proprietary and third-party software, hardware, and consultants highly specialized in certain banking processes. The unit of analysis of this research is the project portfolio of this BU, called *Portfolio Banking*. It has four ongoing projects, all focused on customizing the software of the company, according to the customer's needs, but considering the amount of work done and yet to be done, we considered only three projects, since almost professionals of the development team were allocated to these projects between October 2012 and December 2013, period when the BU adopted the Scrum methodology.

### 3.4 DEVELOPING A CASE STUDY PROTOCOL

According to Yin (2009, p. 79), “the protocol is a major way of increasing the reliability of case study research and is intended to guide the investigator in carrying out the data collection from a single case”. It has the instrument, and the general rules and procedures the investigators must follow during the research. For Yin (2009), a protocol should have the following structure:

- An overview of the case study project (project objectives and auspices, case study issues, and relevant readings about the topic being investigated);
- Field procedures (presentation of credentials, access to the case study "sites", language pertaining to the protection of human subjects, sources of data, and procedural reminders);
- Case study questions (the specific questions that the case study investigator must keep in mind in collecting data, "table shells" for specific arrays of data, and the potential sources of information for answering each question); and
- A guide for the case study report (outline, format for the data, use and presentation of other documentation, and bibliographical information). (Yin, 2009, p. 81).

The APPENDIX A – CASE STUDY PROTOCOL describes the case study protocol.

### 3.5 COLLECTING EVIDENCES

According to Yin (2009), case study evidence has the advantage of dealing with various types of evidences, such as, documents, archival records, interviews, direct observation, participant-observation, and physical artifacts. He gives attention to three important principles to data collection effort that will increase the quality of the case study, which include the use of:

- multiple sources of evidence (evidence from two or more sources, converging on the same facts or findings);
- a case study database (a formal assembly of evidence distinct from the final case study report); and
- a chain of evidence (explicit links among the questions asked, the data collected, and the conclusions drawn). (Yin, 2009, p. 98)

The following subsections describe in more detail these three principles.

### 3.5.1 Use of Multiple Source of Evidence

For Yin (2009, p. 115-116), “the most important advantage presented by using multiple sources of evidence is the development of converging lines of inquiry, a process of triangulation and corroboration... Any case study finding or conclusion is likely to be more convincing and accurate if it is based on several different sources of information, following a corroboratory mode”. This research uses documents, interviews, and participant-observation for data collection.

#### 3.5.1.1 Documentation

Documentary information is relevant to the research of every case study and this type of information takes many forms, so it is important to have specific plans for data collection (Yin, 2009). Table 10 was constructed to direct the investigation of the documents and their updates, if applicable, previously described in the section 3.1.2.3.

**Table 10: Types of Documents to Be Collected**

<b>Documentation</b>	<b>What kind of data?</b>	<b>Who may provide?</b>
Request for proposals (RFP)	Products and services hired, type of contract (e.g.: fixed-price, time-and-material, etc.), customer duties, etc.	Project Manager
Proposals	Products and services offered according to the RFP.	Project Manager
Contracts	Products and services hired.	Project Manager
Project plans	The overall main project management plan and risk management plan.	Project Manager
Schedules	The initial schedule proposed and how the updated schedules evolved during the project execution.	Project Manager
Technical and functional specs	The original scope of the projects and changes in the scope.	Project Manager, Development Manager
Reports	Data and information distributed to the stakeholders.	Project Manager
E-mails	Relevant information exchanged between the company and customer, mainly related to the scope and schedule.	Everyone
Other project documents	Additional data, if applicable.	Everyone

Source: created by the author.

Table 10 illustrates what types of documents were collected, what kind of information was found in these documents, and who provided them, according to the list of people interviewed.

### 3.5.1.2 Interviews

According to Yin (2009, p. 108), “interviews are an essential source of case study evidence because most case studies are about human affairs or behavioral events. Well-informed interviewees can provide important insights into such affairs or events. The interviewees also can provide shortcuts to the prior history of such situations, helping you to identify other relevant sources of evidence”.

There are two common types of interview: in-depth interviews and focused-interviews. In a focused-interview, a case study protocol is used to guide the inquiry (Yin, 2009). This study used a case study protocol (see APPENDIX A – CASE STUDY PROTOCOL) to support the interview of the following employees:

- Development manager;
- Project managers of each project in the portfolio; and
- Two key professionals of the software development team.

Interviews were carried out from November to December 2013. Each interview was taped (a total of 8.42 hours) and transcribed verbally in a structure sheet composed of all propositions and respective questions. This model facilitated the writing of the findings once it was easy to compare the respondents’ answers. Notes were also taken for some questions that rose after the interviews and the writing were incorporated to the findings.

### 3.5.1.3 Participant-observation

Participant-observation is a type of data-collection where the investigator has a specific role in the case. It provides some opportunities for collecting case study evidences and involves some potential problems. The greatest opportunities are to gain access to events or people that are otherwise inaccessible to a study, and to perceive the reality of the case study from an inside view-point. The major problems are the potential biases produced, since the investigator has more difficulty to work as an external observer, the risk of the investigator becoming a supporter of the

company being studied, and the time consumed by the professional to the detriment of the investigator (Yin, 2009).

The author of this study has a formal management position in the company being studied and stayed alert to all major problems described above, using all research instruments defined in this study to minimize the risk of invalidating the credibility of the research. To reduce the bias in the study, he chosen does not express his own opinion on the issues raised by respondents, but just use his knowledge about the company to bring relevant information that could be evidenced.

### 3.5.2 Use of a Case Study Database

According to Yin (2009, p. 119), “every case study project should strive to develop a formal, presentable database, so that in principle, other investigators can review the evidence directly and not be limited to the written case study reports. In this manner, a case study database markedly increases the reliability of the entire case study”.

Considering the time constraint to develop this research and the confidentiality of all data and information provided by the company, it was not the focus of this study to create an explicit case study database, therefore all relevant case study evidences is presented in the analysis and discussion chapter.

### 3.5.3 Use of a Chain of Evidence

The use of a chain of evidence is to allow an external observer to follow the derivation of any evidence from initial research questions to case study conclusions or the opposite, from the conclusions to the questions (Yin, 2009). This study uses a set of strategies and methodological approaches to link the questions to the conclusions, including the formulation of theoretical proposition, the construction of a construct, and the development of a case study protocol.



### 3.6 ANALYSING THE CASE STUDY EVIDENCE

According to Yin (2009, p. 126), “data analysis consists of examining, categorizing, tabulating, testing, or otherwise recombining evidence, to draw empirically based conclusions”. He argues that is fundamental to have an analytic strategy to guide the correct and fair data analysis, and to make manipulations more effectively and efficiently. Four strategies are suggested by him:

- relying on theoretical propositions (the most preferred strategy);
- developing case descriptions;
- using both quantitative and qualitative data; and
- examining rival explanation.

Five analytic techniques for analyzing case studies can be used with one of these strategies:

- pattern matching;
- explanation building;
- time-serious analysis;
- logic models; and
- cross-case synthesis (specific to analysis of multiple cases).

Yin (2009) argues that four principles underlie all good social science research, this is, the analysis should i) demonstrate that attention was given to all evidences; ii) address, if possible, all major rival interpretations; iii) address the most significant aspect of the case; and iv) use the investigator own prior, expert knowledge in the case study.

This research is based on i) relying on theoretical proposition strategy, since the propositions supported the research question and the research objectives; and ii) pattern matching analytic technique, in which the logic compares an empirically based pattern with a predicted one. For Yin (2009, p. 136), “if the patterns coincide, the results can help a case study to strengthen its internal validity”.

### 3.7 REPORTING THE RESULTS

The results are reported in chapters 4 and 5. Chapter 4 describes the analysis and discussion on the case study, including the following sections:

1. Context of the case studied (organizational structure, responsibilities, challenges, and limitations of the project managers, and the unit of analysis);
2. Agile software development practices identified in the case;
3. PPM practices identified in the case;
4. Types of uncertainties identified in the portfolio and practices to handle them;

Chapter 5 presented the conclusion summarizing the findings, the contributions for academia and practice, the limitations of the study, and suggestions for future works.

### 3.8 CRITERIA FOR JUDGING THE QUALITY OF RESEARCH DESIGNS

According to Yin (2009), there are four tests commonly used to judge the quality of any empirical social research: i) construct validity; ii) internal validity; iii) external validity; and iv) reliability. Table 11 shows it. As a case study is an empirical study, it is possible to use these tests to evaluate the research design proposed.

**Table 11: Case Study Tactics for Four Design Tests**

Tests	Case Study Tactic	Phase of research in which tactic occurs
Construct validity	<ul style="list-style-type: none"> <li>• use multiple sources of evidence</li> <li>• establish chain of evidence</li> <li>• have key informants review draft case study report</li> </ul>	data collection data collection composition
Internal validity	<ul style="list-style-type: none"> <li>• do pattern matching</li> <li>• do explanation building</li> <li>• address rival explanations</li> <li>• use logic models</li> </ul>	data analysis data analysis data analysis data analysis
External validity	<ul style="list-style-type: none"> <li>• use theory in single-case studies</li> <li>• use replication logic in multiple-case studies</li> </ul>	research design research design
Reliability	<ul style="list-style-type: none"> <li>• use case study protocol</li> <li>• develop case study database</li> </ul>	data collection data collection

Source: adapted from Yin (2009, p. 41).

Table 11 summarizes the four tests, the recommended case study tactics, and the phase of research when the use of this tactic is suggested.

### 3.8.1 Validity

The validity test is related to the capacity of the instrument to measure *de facto* what it proposed to measure, and is divided into construct validity, internal validity, and external validity (Martins & Theóphilo, 2009; Yin, 2009).

#### 3.8.1.1 Construct Validity

The construct validity refers how well the construct reflects the theoretical means (Martins & Theóphilo, 2009). Yin (2009) presents three common case study tactics to increase construct validity: use of multiple sources of evidence, establish a chain of evidence, and have the draft case study report reviewed by key informants. The first two tactics are used in this research, as described in the section 3.5.

#### 3.8.1.2 Internal Validity

The internal validity seeks to establish a causal relationship, whereby certain conditions can lead to other conditions. It is indicated for explanatory case studies, not for exploratory or descriptive case studies (Yin, 2009). As this is an exploratory case study, this research does not focus on internal validity.

#### 3.8.1.3 External Validity

The external validity focuses on discovering if the study's findings are generalizable beyond the immediate case study. Case study relies on analytic generalization, in which, the investigator is trying to generalize a particular set of results to some broader theory (Martins & Theóphilo, 2009; Yin, 2009). This research described the context and environment of the case studied in chapter 4, considering the project uncertainty and complexity, to give subsidy to future researchers.

### 3.8.2 Reliability

The reliability test is related to the constancy of results when the same individual or object is measured more than once. Validity and reliability are applicable to measures derived from tests, instruments of data collection, measurements techniques, and the research itself (Martins & Theóphilo, 2009). Yin (2009) presents two common case study tactics to increase reliability: use case study protocol and develop case study database. This study followed the primer tactics, creating a research protocol, as described in the section 3.4.

## 4 ANALYSIS AND DISCUSSION

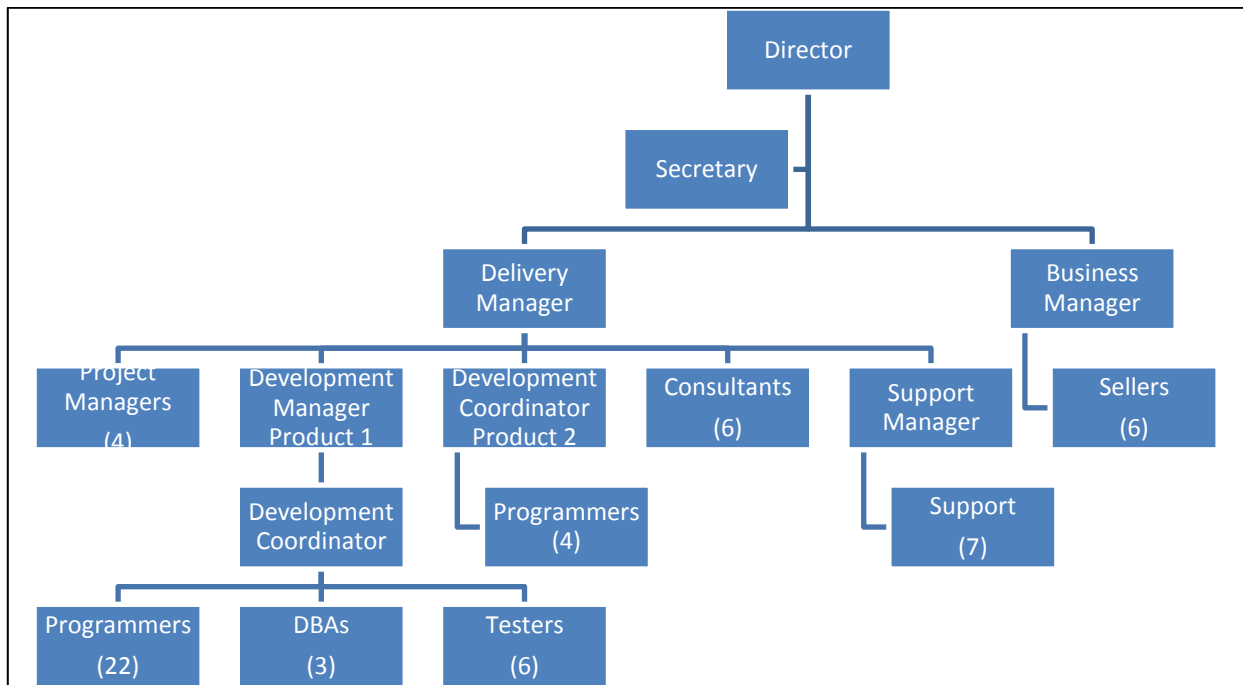
This section presents the context of the case studied (organizational structure, responsibilities, challenges, and limitations of the project managers, description of the software solution, and the unit of analysis), the agile software development practices identified in the case, the PPM practices identified in the case, and the types of uncertainties identified in the portfolio and practices to handle them.

### 4.1 ORGANIZATIONAL CONTEXT

The organization nicknamed *Company Brazil* is a subsidiary of a multinational company dedicated to the research, development, and integration of technological solutions, including software and hardware. It has three BUs, all focused on providing custom services. One of these BUs is focused on the financial sector – banking. This BU offers integrated solutions that include customization and integration of proprietary and third-party software, hardware, and consultants highly specialized in certain banking processes. There is intensive use of skilled labor to provide cutting edge solutions.

#### 4.1.1 Organizational Structure

Figure 16 illustrates the organizational structure of this BU and, for each level, the number of professionals working at the time of the investigation, if there is more than one professional. APPENDIX B - DEVELOPMENT TEAM MEMBER OF THE PRODUCT 1 shows an overview of the BU's professionals under the development manager of the product 1. This product is the same described in section 4.1.3.



**Figure 16: Organizational Structure of the Banking Business Unit**

Source: created by the author.

The delivery manager must be able to have a holistic vision of the actual and future demands, and the company's national and international strategies according to the orientation of the BU's director. Project managers have their project needs that are exposed to the delivery and development managers. Considering all these information, the delivery manager defines and gives the main orientations for the development manager to prioritize the execution of the demands. The development manager shares this responsibility with the development coordinator who organizes and allocates the resources according to the demands of the projects, always validated by the development manager.

The BU has three groups of professionals in the development team of the product 1: programmers, database administrators (DBAs), and testers. A small part of the team is composed by senior professionals and the largest part, by juniors and trainees. Some project managers and key specialists interviewed understand that there are no intermediate professionals, just seniors, juniors or trainees. The programmers are divided in two groups: C/C++ group (18 programmers) and Java group (04 programmers), but all professionals work constantly together.

#### 4.1.2 Responsibilities, Challenges, and Limitations of the Project Managers

The three project managers interviewed have from 11 to 30 years of experience in global companies working on IT projects, and performing a managerial role in projects of software development, integration, customization, infrastructure, and telecommunication. They are PMI's Project Management Professional (PMP) certified and almost all their experience has been focused on traditional project management. In the last year they are experimenting, in different degrees, the Scrum methodology according to each project managed. One of the project managers is also the development manager.

The main responsibilities and challenges of their position are achieving the success of the projects, which means meeting the customer's needs, and the financial results (revenue, cash flow, and billing). They also have to support the pre-sales, manage the customer relationship, and explore new business opportunities into their projects.

They complained that there is not a clear deadline to finish the project activities and to start the support activities, common in this type of project. In practice, all project managers are also responsible for giving support to their customers during the project execution. So they have to manage the pre-sales, the project, and the support. As stated by one project manager, "this company has the most complete demand for a project manager I have ever seen in the market". Other challenge is to sensitize the company's commercial area and customers that a more honest and correct project deadline is healthier for everyone, avoiding stress and wear, given the complexity of the solution that has been agreed.

The project managers understand that the BU has a weak matrix structure, so they do not have functional or hierarchical authority to define and allocate resources (human, financial, and infrastructure). All project managers have to share the same resources so, for example, it is very common to find human resources that have worked in all projects. On the other hand, considering that the type of the projects launched by the BU must have a relevant number of senior professionals, they agree that this structure is better to share costs and to optimize resource allocation.

For them, the problem of this sharing is the impact on time, commonly affected taking into account the aggressive deadline of the projects. Some project managers understand that the

BU must to evolve to a strong matrix structure because the current one does not fulfill all benefits of cost and resources sharing, productivity, and resources retention.

Another limitation of the project managers is related to some financial decisions that affect the project margin. They find varied opportunities within the projects to generate new revenues such as requesting of new requirements or changes, new business needs, but they do not have the autonomy to decide if these opportunities can be charged. There are evidences of some features that were built into the solution as changes, but were not paid by the customers. There are still costs of presales not related to the project that are embedded into the project costs, and the project managers have to accept it.

#### 4.1.3 Description of the Software Solution

The BU has one product that has been evolving for more than 12 years. It is a software to capture and process financial and non-financial documents, such as, checks, money orders, deposit slips, and payment coupons, supporting the banking's back-office. There is intensive customization of this software to create, modify, remove, or enhance its functional and non-functional features according to customer's business needs. It also includes integration between this software, customer back-end applications, and third-party software.

This software is a client-server solution and has more than 73 client components, 32 web components, and 41 server components. Part of these is considered core components, so any change may affect all the solution. The other parts can be changed with or without minimal interference on other components. Most server and client components are developed in C++, few client components are also developed in Visual Basic, and all web components are developed in Java technology.

#### 4.1.4 Unit of Analysis

The unit of analysis of this research is the BU's portfolio called *Portfolio Banking*. It has four ongoing projects, all focused on customizing the software of the company, according to the customer's needs. Considering the amount of work done and yet to be done, we considered only



three projects, since almost professionals of the development team were allocated to these projects between October 2012 and December 2013, period when the BU adopted the Scrum methodology. The next paragraphs describe in detail each one of these projects.

The first project, nicknamed *Project Alpha*, was purchased by a Brazilian public bank. Their main objective was the acquisition of a solution for the capture and processing of financial documents through image capture. It includes licenses of software for 3041 users, software customization, software installation, technical support, technology updates for 36 months and knowledge transfer.

The second project, nicknamed *Project Beta*, was purchased by another Brazilian public bank. This time the main objective was the acquisition of a solution for the capture and processing of financial, non-financial, and automated conference of formal aspects and signatures, through decentralized image capture and centralized processing. It includes licenses of software for 8700 users, software customization, hardware, software and hardware installation, technical support, 60 months for technology updates, knowledge transfer, and 6000 hours for integration, customization, and training.

The third project, nicknamed *Project Gama*, was purchased by a private bank directly with a business partner of the company studied, thus there is only a contractual relationship between the customer and the business partner, and between him and the company. For the general purposes of this study, the customer of the BU is the final customer, and not the business partner.

The main objective was the acquisition of a solution for capture and processing of financial and non-financial documents through image capture. It includes licenses of software for processing 150 millions of documents per year, software customization, software installation, technical support, and technology updates for 60 months. Differently of the projects Alpha and Beta, this acquisition by the customer has a significant strategic goal that is the substitution of the current and similar solution running in the bank for those purposes.

Table 12 summarizes some information collected in interviews, documents such as RFP, contracts, and project documents, and according to the scoring model proposed by Little (2005), these projects can use some agile methodology to handle uncertainties, as a result of the evaluation of the complexity and impact, considering the Houston Matrix Quadrant Assessment.

**Table 12: Main Characteristics of Projects**

Information	Project		
	Alpha	Beta	Gama
Customer	Public bank		Private bank
Date of Start	August 2009	November 2010	January 2013
Date of Signature	August 2009	December 2010	August 2013
Type of contract	Fixed price	Fixed price	Fixed price
First release deployed into SIT	February 2010	March 2011	April 2013
First release deployed into production	March 2011	May 2011	March 2014
Ongoing Customization	Yes		
Mission criticality	Mission-critical with large user base		
Team location	Same time zone +/- 2 hrs.		
Team capacity	Team with limited experience and a few experts		
Domain knowledge gaps	Developers have no idea about the domain	Developers require some domain assistance	Developers have exposure to the domain
Dependencies	Significant		
Team size	5	15	40
Complexity	29	27	34
Market uncertainty	New, unknown, and untested market	Initial guess of market target likely to require steering	
Technical uncertainty	We're not quite sure if we know how to build it	We think we know how to build it	
Project duration	24 months		
Dependencies, scope flexibility	Some published interfaces Scope is highly flexible		
Uncertainty	12	12	12
Houston Matrix Quadrant Assessment	Bulls		

Source: created by the author.

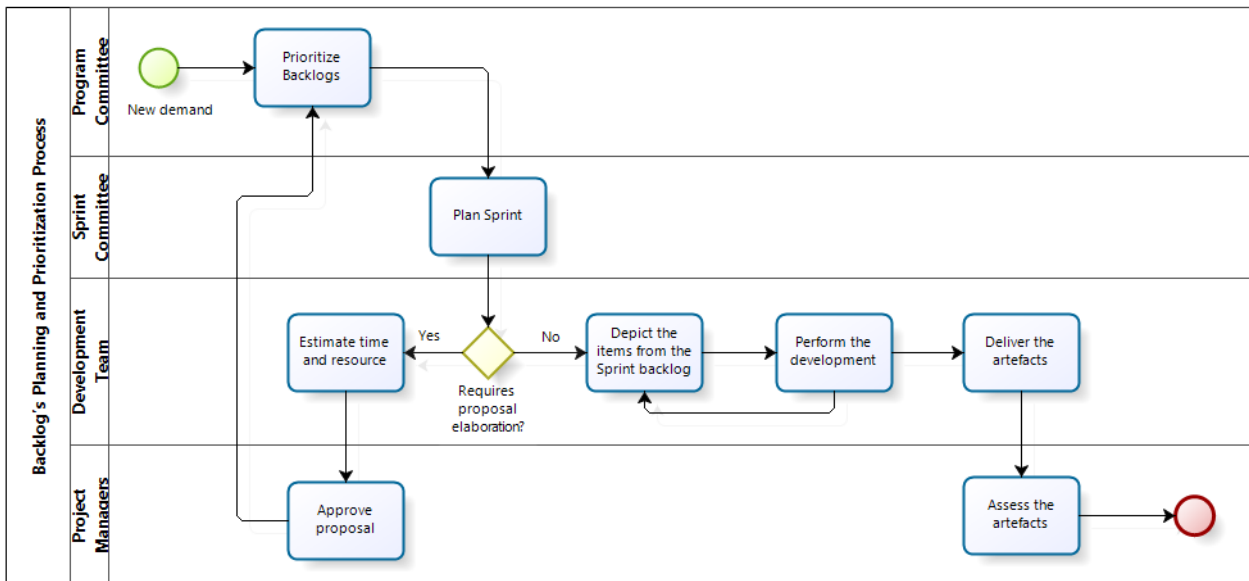
## 4.2 AGILE SOFTWARE DEVELOPMENT PRACTICES

According to all professionals interviewed, and informal conversations with the development team, there was not any kind of software development process in the BU until mid-2012. All project managers were directly seeking for professionals from the development team to attend their project's needs. It was a tough challenge for each professional to balance the demands because there is not a clear definition of priorities. In practice, they did for those that

“shouted” first and louder. The development coordinator tried to get involved as he gained the trust of the development team, since he had been in this role for only five months. The scenario was very chaotic and stressful for everyone affecting the deadlines and quality of the delivery.

Trying to figure out a way to change this situation, the development team decided to adopt some Scrum methodology practices. They received the support of almost all managers in the delivery team, and started holding informal meetings to plan the Sprints, and Daily Scrum to monitor the evolution of the work in progress. There is no evidence that this was done frequently until October 2012.

After that, the development team and project managers decided to implement a formal process to plan and prioritize the project’s backlog, following more in depth the Scrum practices adapted to the BU’s context, as suggested by Imbrizi and Maccari (2013). Four roles were defined for this process: Program Committee, Sprint Committee, Development Team, and Project Managers. Figure 17 illustrates these roles and their main responsibilities within the process.



**Figure 17: Backlog’s Planning and Prioritization Process**

Source: created by the author.

The Program Committee is composed by the project managers, the development manager (such as the Product Owner in Scrum), and the development coordinator (such as the Scrum Master in Scrum). They are responsible for selecting the product backlog items of all projects that

will be prioritized into the next Sprint. All project managers expose their own demands and priorities. In case of conflicts, the final decision is the responsibility of the development manager who can request the opinion of the delivery manager.

The Sprint Committee is composed by the development coordinator, the chief and web architects, the DBAs, and other key professionals of the development team. They are responsible for evaluating all items prioritized by the program committee and confirm if there are enough resources and time to do all the work. In case of overload, they have to inform the project managers those items that will not be given attention in the next Sprint. They also have to breakdown the items into small parts to distribute the tasks for the development team.

The Development Team is composed by programmers, DBAs, and testers. They are responsible for performing the development of the backlog's items. It includes codifying, testing, and/or evaluation of work efforts for new features, enhancements, and bug fixes. They also have to depict the items from the Sprint backlog into small items to define the tasks to be done.

The Project Managers refers to all the project managers in the BU. As described in more detail in the section 4.1.2, they are responsible for achieving the success of the projects. They have to evaluate the work efforts for new demands and the artifacts delivered by the development team suggesting some adjustments if necessary. Table 13 resumes these teams and their main responsibilities.

**Table 13: Main Roles of the Backlog's Planning and Prioritization Process**

<b>Team</b>	<b>Participants</b>	<b>Main responsibilities</b>
Program Committee	Project managers, development manager, and development coordinator.	Select the product backlog items of all projects that will be prioritized.
Sprint Committee	Development coordinator, chief and web architects, DBAs, and others key professionals of the development team.	Evaluate all items prioritized and confirm if there are enough resources and time to do all the work.
Development Team	Programmers, DBAs, and testers.	Perform the development and tests of the new features, enhancements, and bug fixes.
Project Managers	All project managers of the BU.	Deliver the artifacts to the customer and give the results of the acceptance tests.

Source: created by the author.

This formal process was followed in the first two months, but after that there is no evidence that the two committees met regularly. Although the professionals interviewed agreed they stopped the formal meetings, the Sprint planning was kept by the development coordinator

which gathered all demands individually and directly with the project managers, and aligned with the development manager the items to be prioritized. A set of reports generated and published for every Sprint since November 2012 on BU's intranet are evidence of it. APPENDIX C – SPRINT REPORTING TEMPLATE shows one sample of these reports.

The project coordinator uses these reports to manage and control the development team allocation. All activities related to issues on customer production have higher priority than issues related to new features or enhancements because the BU adheres to rigorous service level agreement (SLA) according to terms in contracts.

A white board and sticky notes are used to expose the team's activities reported. This board is divided into three groups of activities (in progress, test, done) for each project described before. Each note is divided into three parts (description of the activity, who is performing the activity, and who is testing the artifact, if applicable). Each professional of the development team is responsible for writing out his/her name on the note. The project coordinator believes this is a way of giving accountability to each team member.

#### 4.2.1 Organization and Interactions of People

The respondents said that in the beginning of the Scrum implementation the development team was wary and suspicious, and the organizational climate was quite negative. Sometimes they complained about the Daily Scrums. Senior professionals were afraid of assuming more duties of technical definition because the organizational history shows that the chief architect has concentrated all software's definitions once he is the main solution expert.

After some Sprints and Daily Scrums the dissemination of knowledge and the communication between the team's members improved greatly. The senior professionals have taken more responsibility and risk for their actions regardless of availability of the chief architect. As said by one project manager, "they broke a barrier that everything depends on one person".

All development team members were exposed during the Daily Scrum which helped to show who was hard working and who was not. Some members left the team, others were exchanged. The respondents believe that the organizational climate is much better now. It's easy to find a friendly atmosphere and banter between colleagues not seen in the past.

The development team is beginning to trust in this agile methodology, being more responsible, dedicated, and aware of their duties and how this affects the project's goals. If the team members do not know what to do they ask and chase the solution. Everyone is trying to talk with each other finding ways to achieve a better result. One project manager said that "the agile development process brought evolution and maturity to the team".

The respondents understand that there is a natural technical knowledge gap, which cause is the lack of experience of the development team since 20 members are trainees and juniors (64.5% of the team). Some project managers said that another gap is related to a better dissemination of the agile methodology knowledge by the development coordinator. They agreed that some team members believe that this is a particular reserve of the coordinator and not a practice widely adopted by the market. So, they suggest a better explanation of agile methodology by comparing it with other traditional practices.

#### 4.2.2 Software Releases and Assessment

The next subsections describe the findings of versioning, building, releasing, range of release, and the evaluation of software release.

##### 4.2.2.1 Versioning, Building, and Releasing

The processes of versioning, building, and releasing of software in this BU are historically chaotic. The investigation showed that at least until the beginnings of 2013, there was not any kind of formal process to deal with them for all projects. The BU has a SCM repository and building servers for a long time, but just the chief architect and one or two developers had the understanding, but not the complete control over them.

The lack of knowledge and processes often resulted in bugs caused by one developer's code stepping on another developer's code, developers working over wrong codes, loss of source code, packaging of incomplete features, deployment of wrong binaries, re-emergence of bugs that had been fixed, and so on. There are evidences of some initiatives that had tried resolving these damaging problems, but all failed because of a lack of leadership and engagement of the whole development team.

After April 2013, the chief architect took for himself the responsibility to organize, restructure, control, monitor, and communicate the new process to use the SCM repository. A clear documentation was created to demonstrate how to operate the repository. After few weeks, this architect trained all members of the development team to reinforce the correct use of the repository. Some changes were made according to the new findings during the working performing to better meet the team's needs.

After this initiative, other senior professionals engaged to implement a process to automate the building every day (nightly build process) to verify if the source-code is still healthy and not broken. It is good to find the following problems: checked in code that breaks other code, programmers forget to check in a necessary file, build script stopped working, building machine stopped building, etc. Other initiative included the use of tools to monitor and alert any change into the repository.

The releasing process was also changed to maximize the success of the software installation and maintenance. It was implemented the Red Hat Package Manager or RPM Package Manager (RPM), a package management system which has the advantages of simplicity, consistency, and automation, compared to a manual building. This was another initiative of a senior developer.

Although these actions have been taken, only the project Gama has been benefited from all these initiatives. Project Beta project manager recognizes that these initiatives are in progress but he cannot perceive benefits in his project. The investigation confirmed that the building and releasing processes implemented above were not applied in his project yet.

#### 4.2.2.2 Range of Release

The respondents said that the frequency of delivery should vary according to the project's contract. Each contract has definitions related to the type of delivery such as problems on production, bugs, new features, enhancements, and changes. For each type there is a specific deadline for completion. The original deadlines of each project show varied month periods to deliver the first features, and few hours or days for resolution of bugs and problems on production. Table 14 presents the estimated time for each type of delivery in the projects.

**Table 14: Estimated Time for Each Type of Delivery**

Type of delivery	Estimated time for each project (Initial Response/Resolution)		
	Alpha	Beta	Gama
First features to deliver in SIT	2 months	6 months	3 months
Severity 1 – critical error	15min / 4h	30min / 2h	1h/8h
Severity 2 – significant error	2h / 8h	2h / 4h	2h/10h
Severity 3 – moderate error	4h / 24h		
Severity 4 – minor error	24h / 40h	24h / 48h	24h/30h
New features or enhancements	Under estimation		

Source: created by the author.

In practice, the real scenario is quite different. A lot of uncertainties affect the project and contribute to change the actual deadlines achieved by the development team. The types of uncertainties will be discussed in section 4.4. As a result of these changes, it is very common to find shorter and longer deadlines for original features and longer for bug fixes. There are evidences, for example, of deadlines that were advanced by months to deliver original features, and bugs in the backlog for months waiting a moment to be fixed. It is the opposite of the expected result. In fact, all projects have faced damaging delays.

The BU decided to adopt a Sprint of two weeks because considering the context of its projects, one week is a very short-term to deliver new feature or enhancements, and three weeks is a very long-term once changes happen with high frequency as will be discussed in section 4.4. Consequently, project managers and the development team try to adapt their planning to this period of time – two weeks. Into this period they are expected to generate as many software packages of new functionalities, enhancements, and bug fixes, as necessary according to the demands and urgency of each project.

#### 4.2.2.3 Evaluation of the Software Release

All projects have basically two phases to evaluate the software release: system integration testing (SIT) and user acceptance testing (UAT). In the SIT phase, professionals of the BU test all functional and non-functional features released into a development environment of the customer.



In the UAT phase, only customer's professionals do it in an acceptance environment of the customer.

The installation and configuration of these environments were done according to each customer's policy. At projects Alpha and Beta, BU's professionals installed and configured both the development and acceptance environments. At project Gama, the BU's professionals installed and configured the development environment, and the customer's professionals do it in the acceptance environment, but with the support of BU's professionals.

How each customer evaluates the release varies in different ways. At projects Alpha and Beta there is no formal process to assess the release. The customers of these projects test the software using just the requirements as a reference, but without a specific documentation such as test cases to guide them. On the other hand, at project Gama, the customer does it using an extensive set of test cases all registered into a quality assurance (QA) system.

The UAT at project Beta is a little complex to manage because the customer has business and IT professionals sharing the responsibility to evaluate the software, but they have different interests, are not in the same hierarchical structure, and do not follow the same line of organization, operation, and evaluation. To make things worse, the business professionals have a lack of knowledge in requirement analysis, testing, and result analysis.

Before the software release, an internal testing should be done, when testers of the development team test all functional and non-functional features into the testing environment of the BU. The evidences showed that it is not what really happens for all projects. There are no formal processes to test the software, but only isolated initiatives of some professionals. In some cases, the project manager and the development coordinator were the testers. In fact, historically the BU never had a group of testers focus on testing. The developers used to do it.

Recently, there were two initiatives to try to solve this problem. Trainees were hired and some professionals from the support team were transferred to focus on testing either into the testing environment of the BU or into the SIT. The last initiative came from a project manager to aide his own project, since he had previous experience with the support team.

### 4.2.3 Customer Collaboration and Development Team Communication

The next sections describe the interactions among the development and the business team, and the customer; the process to communicate changes and request new requirements; market's needs and scope definition.

#### 4.2.3.1 Interaction among Development Team, Business Team, and Customer

The interactions among the development and the business team, and the customer vary in different ways for each project. At project Beta, for example, there was one business analyst for just two month in the beginning of the project, but after that, the project manager took over that role. The project manager reads, understands, criticize, and reviews the customer's requirements before he sends the demands to the development team, usually to the development coordinator and chief architect.

He also avoids the contact of developers with the customer because he believes they do not have time to give all necessary attention to these activities considering the high backlog. The project manager considers that understanding all the requirements he can be prepared for any question and/or complaint of the customer, and better support the development team. He said that takes this role because there are no business analysts in the company to do it.

At project Gama, there is one business analyst from the beginning of the project, who is responsible to interact with the business partner and the customer, and supports the development team to understand the requirements created by the business partner and validated by him. With few exceptions, there is no interaction between the development team and the customer. The Gama's project manager said that initially, just few developers talked with the business analyst, but now everyone in the development team has this opportunity and the project coordinator incentivates this.

#### 4.2.3.2 Process to Communicate Changes and Request New Requirements

The process to communicate change and request new requirements also varies for each project. At project Alpha, the customer sends just e-mails to the project manager which meets with the development team to analyze the requests and estimate time and resources to implement

the new demands. With all these information, the project manager elaborates a proposal and sends it to the customer.

At project Beta, the customer sends a specification of new requirements to the project manager which does all actions defined in the last section, and the development team estimates time and resources to implement the new demands. With all these information, the project manager elaborates a proposal and sends it to the customer. There is a web portal used to include all these demands. This process was implemented one year ago and there is no evidence of any kind of organized process before that, just an extensive e-mail exchange between people.

At project Gama, it is very common to find issues in SIT that are supposed to be bugs, but after the development team's investigation, it has been concluded that legacy system do not work as specified. In this case, the development team informs the business analyst who notifies the business partner about the changes. There are situations in SIT or UAT that the customer disagrees with the product delivered, even if it is in accordance with the specifications. The main cause of this problem is the fact that the business partner has the duty to validate the specification against the customer, but this was not done and the customer also avoided to do this. At the end, the business partner sends new versions of the specification and the BU sends back to him a new proposal with the changes.

#### 4.2.3.3 Market's Needs and Scope Definition

The project managers said that market's needs and scope definition are not well understood and defined. One project manager suggested that only 60% of the scope is known, but the 40% unknown represents 60% or more of the time consumed. The BU's team responsible for creating the solution to the market has a macro view of the banking processes related to their solutions. It is possible to identify in the RFPs published in the market the common and macro needs of all banks. But the business and technical details of these processes are not present in these documents because the banks do not have them well understood, defined or are not published for confidentiality and security reasons. Thus, the complete definition of the project's scope is made only during the project execution.

The projects pose challenges in different ways. At project Alpha, plenty of specifications were developed by the bank's business professionals in the beginning of the project, but there are no evidences of new versions of this specification and it is known that there have been changes.

On the customer side there is a clear responsible for the project that knows in depth all banking processes and has been dedicated to the project from the beginning. He is also the customer's project manager.

At project Beta, the business professionals do not have any expertise in requirement analysis and specification writing, so the end users do not know how to ask or what they want. The project manager said that this is changing because the bank's IT specialists are getting involved in requirement analysis and specification writing.

At project Gama, the scope definition is the worst compared with the other projects because considering the contracting model, the customer was expecting not to be involved in scope definition since he hired the business partner who was supposed to have detailed knowledge of their business processes and one specific application. In practice, the customer perceived that the business partner does not have this knowledge, and he had to allocate shared resources not totally engaged to collaborate with the scope definition at the last minute. There are evidences of dozens of versions of the same specification document. This was still happening at the time this thesis was being written.

#### 4.2.4 Planning and Adapting to Changes

The respondents agree that the company and, consequently, the BU do not have a culture where planning is considered. This comes from the direction and permeates all levels of management. For them, the company is reactive, opportunistic, and driven by commercial decisions. Historically, the company has changed many times over the years and, as one project manager said, "changes are in the DNA of the company".

The BU is not prepared to anticipate customer's needs, but only to react to explicit internal or external customer's demands. BU captures business from opportunities that arise from the market and not as a consequence of long-term planning. All commercial decisions drive the priorities of current and prospective projects, considering the opportunities to sell and bill.

### 4.3 PPM PRACTICES

The next sections describe the portfolio of projects and processes of PPM present in the BU and how its resources are allocated.

#### 4.3.1 Portfolio of Projects

The BU's project portfolio was detailed in section 4.1.4. The portfolio contains projects that have similar proposals for their customers and use basically the same software solution that is customized meeting the specificities of the customer's business processes. They are considered critical to customers' business because if the product resulting from these projects stops, part of their business also stop affecting the financial results.

The respondents said that there is a high dependency between the projects. This dependency can be positive and negative. It is positive when last projects can reuse part of the platform, infrastructure, database, features, enhancements, and documentation of the software delivered by previous projects. They also take advantage of all knowledge acquired and lessons learned.

It is negative when there is parallelism of development, and conflict of resource allocation since the same development team resources can work in all projects. One consequence of this is the overworking to try to minimize the impact on time in all projects. This results in labor liabilities in terms of overtime, and low quality of life for employees who work under high pressure and aggressive deadlines. A sample of 12 professionals showed an average of 160 hours bank of overtime accumulated over the last six months. Other problem identified with the concurrency of resource is resource reallocation when issues arise in SIT, UAT, or production and is necessary immediate action affecting the current planning.

### 4.3.2 PPM Processes

The company has a formal methodology to manage a portfolio of projects, but it is not effectively used at all. One project manager said that; “if the methodology were applied *ipsis litteris*, it would allow for much greater transparency within the project and would prevent us from suffering as we suffer”. Another project manager does not realize that BU has a portfolio management itself because there is no division focused on evolution and innovation of current and new products, neither a commercial team who knows well where BU’s customers are going.

Follow the main processes of project portfolio management suggested by the literature review, the next subsections describe how the BU handles these processes according to the project managers point of view.

#### 4.3.2.1 Identification and Classification

Since the organization is an IS provider, the main criteria used to assess the viability of a project are: i) the technical aspects, i.e., if the company is able to run the project, and ii) the financial risks of the project, as severity of penalties, negative cash flow and expected revenue versus potential losses.

The organization does not rigorously establish, for example, a minimum value for the revenue or margin of the new project. Although these criteria are evaluated, others are considered, such as the business potential and strategic importance of the project for the BU. It is possible that a potential project becomes interesting for the organization, even with considerable financial risk, and without prior planning for the type of project, because of a commercial decision making.

Any process of classification of projects is known, not even similar mechanisms that influence decision making. The projects currently running and prospective projects in general have similar characteristics, varying only the scope that each addresses, given the solutions provided by the company.

#### 4.3.2.2 Evaluation and Selection

For one project manager, the previous experiences and ongoing projects are considered in the evaluation of potential projects. It is possible that a prospect will not go ahead due to a negative environment on ongoing projects. This can happen when managers identify that the same negative scenario can be replicated in new projects.

Another project manager said that if the company is in a healthy and comfortable financial situation, projects of a specific margin and into the BU's market niche are considered, otherwise the BU ventures further into new niches and takes up different margins of the organizational policies. The decision makers are the director, the delivery manager, and the business manager, with the support of the project managers, sellers, pre-sales consultants, and the chief and web architects.

Once a business proposal is sent to the prospective customer and he approves it, the new project will be executed. Although the availability of human, financial and technical resources can be considered in the evaluation and selection process of the projects, the unavailability of them does not mean that the project is not viable.

#### 4.3.2.3 Prioritization

Since the company sells projects and has to deliver them according to contractual conditions, it does not make sense prioritizing a project over another, therefore all projects have to be executed in parallel. What can happen is that it may be necessary to concentrate efforts of human resources in a short period of time, particularly those shared across projects, to meet an urgent demand for an ongoing project. This occurs especially in scenarios that require changes in core components of the company's product.

Project managers identified the main criteria the organization uses to prioritize ongoing projects. They are: i) the strategic importance of the project to the organization, ii) the financial impact of the project on the organization, such as short and long term revenue generation, iii) undesirable costs, like potential penalties, vi) risks of damaging image that may affect the customer relationships, and v) technical aspects.

Although these criteria are known, there is no formal process for evaluating them to make a decision. As the internal pressure of the organization and/or customer increases, it is necessary to take actions to meet the stakeholders' needs. Sometimes the BU accepts taking some risk of

penalties not delivering on time for one customer in the benefit of another if there is a good relationship with the primer customer and financial reasons to do this. As stated in section 4.2.4, the company is reactive, opportunistic, and driven by commercial decisions, according to the respondents.

#### 4.3.2.4 Balancing Portfolio

The resources planning, especially human resources, is done considering the projects with contracts already signed. There are no initiatives for anticipated hiring before the closure of a contract. Each project in progress is organized individually according to a backlog without considering the other projects. The backlog is composed of issues that can be a request of information, enhancement, new feature, or bug fixes.

The BU has a backlog management system to record and monitor all issues but just one project uses it effectively. A project manager has no explicit knowledge of the backlog of another project.

#### 4.3.2.5 Authorization, Review, and Strategic Changes

The authorization is explicit in the act of signing the contract, since the contract project will run. Sometimes, it is possible that a project begins, even without the immediate contract signature. In these cases, there is always a business letter of intent between the parties that sets certain assumptions and constraints, until the contract is signed. This is the case of project Gama who was signed seven months after the beginning of the project.

The company has no policy to stop any project over another. The projects are not canceled because of the high negative financial, legal and image impacts it may cause. The organization has a formal process for the projects monthly review that is conducted under the coordination of a quality area. The main criteria assessed are the financial situation of the project (project scope and billed revenue), schedule and risks.

The BU's strategic planning is reviewed every six months, especially with regard to revenue, billing, and gross margin. These parameters are used as indicators that directly affect the bonus awarded to each employee, in terms of the company's profit sharing plan.



### 4.3.3 Resources Allocation

During the development of proposals the responsible has to suggest the number and type of resources to be allocated and for which period of time. When the business is captured, it is mapped which professionals will be allocated. During the portfolio execution, the resources are allocated to each project into each Sprint planning. Considering that the Sprint at BU is defined as a two week cycle, the resource allocation is visible just in this period of time. After that is difficult to predict where each development team member will be working. The evidences confirm this.

Another factor that affects the resource allocation is the knowledge domain of each professional considering all modules of the BU's product. Today, only few senior professionals can work in almost all the solution components. There are evidences of initiatives to replicate the knowledge of few professionals to the development team members trying to mitigate this constraint that affects all projects' completion.

Before the implementation of Scrum, almost 100% of the resource allocation was planned individually, project by project, without interaction between project managers. As described in the beginning of the section 4.2, the project managers who "shout" louder, seized resources for their projects. There are still situations when an urgent issue, usually in production, forces the reallocation of professionals during the Sprint execution.

The resource reallocation is communicated to the development team, project managers, development manager, and delivery manager. A set of reports is generated and published for every Sprint. APPENDIX C – SPRINT REPORTING TEMPLATE shows one sample of these reports.

Today, the development team likes to know what they are doing and prefer to be involved. They want to know where the BU is going, even if it is to embrace changes, because they wonder what they will do in the coming weeks. According to one project manager, the software development coordinator is trying to balance, with some success and failure, the planned demands of new features and enhancements with urgent issues. The coordinator agrees with this affirmation.

#### 4.4 TYPES OF UNCERTAINTIES AND PRACTICES TO HANDLE THEM

The next subsections describe the types and sources of uncertainties identified in the portfolio and their impacts to the projects and portfolio, how the company manages and controls uncertainty, and how the resources and capabilities are adapted to changing environments.

##### 4.4.1 Types and Sources of Uncertainties

Table 15 summarizes the main sources of uncertainties and their impacts to the projects and portfolio, according to the respondents.

**Table 15: Main Types of Uncertainties and their Impacts to the Projects and Portfolio**

<b>Types of uncertainties</b>	<b>Impacts to the projects and portfolio</b>
<ol style="list-style-type: none"> <li>1. The detailed scope is not known;</li> <li>2. Signing the contract after the project has started, which in practice implies assuming all changes as being part of the original scope at the time of the signature, without opportunities to manage the changes;</li> <li>3. The political conflicts between the business and the customer's IT professionals, since it is very difficult to foresee these conflicts before the beginning of the project;</li> <li>4. Last minute requests to meet customer's security and infrastructure policies;</li> <li>5. Business partner without knowledge of customer's business processes and application; and</li> <li>6. The unavailability of the legacy of the customer for testing during the development phase.</li> </ol>	<ol style="list-style-type: none"> <li>1. The development team works overtime, in a stressful environment and not motivated;</li> <li>2. The BU assumes risky deadlines in a specialized and short market;</li> <li>3. There is a much greater amount of time than planned, creating overtime bank, which is a labor liability for the company;</li> <li>4. High cost because of not expected extra hours, travel, and lodging;</li> <li>5. Much more parallelism of work in the project and with other projects, since there scope is larger than planned but with the same deadline;</li> <li>6. The deadlines assumed with the customer are not reached;</li> <li>7. The company's reputation is harmed with the customer;</li> <li>8. Delay in receiving and invoicing;</li> <li>9. Structural changes in the software to meet security and infrastructure policies; and</li> <li>10. The solution is partially tested during the development phase.</li> </ol>

Source: created by the author.

#### 4.4.2 Practices to Manage and Control Uncertainty

The project managers understand that the BU does not manage uncertainties correctly following common practices in the market, although they have years of experience with project management. Rarely there are some actions to manage changes considering scope, time, and cost. They argued that almost all actions are based on trust between the customer's key professionals and the BU's project managers. At same time they have been taking some actions to minimize the impacts of changes presented in the previous section, such as:

- a) Allocating a senior business analyst full time to understand the customer's business processes to minimize the impact of the business partner lack of knowledge;
- b) Delivering parts of the solution at the beginning of the SIT, allowing the team to perform the SIT on the planned date, and delivering the remaining parts during the execution of the SIT, permitting the customer to see the evolving of deliveries, and;
- c) Running the Daily Scrum, it is possible to identify who is having difficulties or problems to perform their work, given the opportunity for the development team to support and/or reallocate resources, if necessary.

#### 4.4.3 Reallocation and Re-optimization of Resources and Capabilities

According to the project managers, the development team members execute what is attributed to them, maybe not in the way they wanted but within the time that is given to them. They could be doing much better if the company had given them the entire necessary infrastructure (equipment, training, management tools, methodology, templates, technical documentation, and lessons learned).

The new professionals entering the BU find it difficult to engage because things are not structured in such a way that they can be easily found without producing. The knowledge is very much in the mind of the senior members of the development team.

There are evidences of some informal initiatives of the development team to better prepare the newcomers that include the following steps:

1. The development coordinator supplies three types of documentation: the scope of one RFP to introduce what kind of solution the market is buying, the technical proposal of the BU related to this RFP, and the most current manuals of the solution;
2. One tester of the development team does a quick presentation of the main features of the company's solution;
3. The chief architect or one senior developer gives a brief explanation of one specific part of the solution, considering that the new professional will work on this part;
4. The new professional is motivated to read and understand a specific piece of code to understand how the software is written with the support of the chief architect or one senior developer; and
5. Pair programming is applied in early development.

As described in the sections before, there are high levels of changes in the projects, thus the development team are constantly changing the focus of one activity to another, especially to solve problems in production, SIT and UAT. A review of the last 24 Sprint reports showed that new unplanned issues, that arose during the Sprint account for 31.72 % of total issues, and only 55.56% of the issues planned during the Sprint Planning were delivered on time. It is clear that part of this result is related to the resource reallocation.

Trying to maximize the time of solution, there are evidences that the BU is mapping for each developer what modules he/she worked. The main objective is to have at least two developers with sufficient knowledge on each module of the solution avoiding the dependency of only one professional and concurrency between projects and within the same project.

All development team members are aware of the high priority of the issues on customer's production, so they are instructed to act immediately in these cases. They are also very proactive to find changes during the delivering, especially in SIT when supposed bugs are identified as differences of specification.

#### 4.5 SUMMARISING

This section presented the context, the agile software development and PPM practices, the types of uncertainties, and practices to handle them identified in the case studied. Table 16 illustrates for each value and principles defined by Beck et al. (2001), and each best practice identified by Leffingwell (2007), if there is evidence, partial evidence, or no evidence of them in the case studied.

It is possible to find out, considering the short time that the BU is implementing the Scrum – one year approximately, that the company is evolving in the process to use the agile methodology to contribute to its PPM, as will be detailed in the next section. Indeed, according to literature review, it was not expected that company had applied all values, principles, and practices of agile in just one time, since it is a process of continuous improvement. Some evidences demonstrated the interest of the company to keep evolving in the implementation of agile.

**Table 16: Values, Principles, and Practices Evidenced and Not Evidenced**

<b>Values</b>	<b>Principles</b>	<b>Practices</b>
Value 01 – evidenced	Principle 01 – evidenced	Practice 01 – partially evidenced
Value 02 – evidenced	Principle 02 – evidenced	Practice 02 – partially evidenced
Value 03 – not evidenced	Principle 03 – evidenced	Practice 03 – partially evidenced
Value 04 - evidenced	Principle 04 – not evidenced	Practice 04 – evidenced
	Principle 05 – evidenced	Practice 05 – not evidenced
	Principle 06 – evidenced	Practice 06 – evidenced
	Principle 07 – evidenced	Practice 07 – not evidenced
	Principle 08 – not evidenced	
	Principle 09 – not evidenced	
	Principle 10 – partially evidenced	
	Principle 11 – partially evidenced	
	Principle 12 – not evidenced	

Source: created by the author.

## 5 FINAL REMARKS

The next sections summarize the conclusion, contributions for academia and practices, limitations of this study and suggestions for future works.

### 5.1 CONCLUSION

This master's research was undertaken to investigate the field of agile software development and PPM in dynamic environments of an integrated solution provider. The objective was to attempt to answer the following research question: *How agile software development contributes to project portfolio management in dynamic environments of an integrated solution provider?* The strategy was to explore it through the qualitative study of one portfolio of three projects in one company – single case study.

As described in the section 4, after the implementation of the agile methodology Scrum, it is possible to identify some benefits that are contributing to the BU's PPM, as follows:

- The resource reallocation between projects is better supported by the information provided from the Sprint reports;
- The Daily Scrum gives the opportunity to anticipate any resource reallocation to avoid or minimize the delay of delivers.
- Senior professionals have taken more responsibility and risk for their actions, getting more engaged to implement new initiatives and make decisions;
- The development team is being more responsible, dedicated, and aware of their duties and how this affects the project's goals.
- The dissemination of knowledge and the communication between team members improved greatly, keeping the whole development team aware of how things are going in all project portfolios.
- The adoption of a white board and sticky notes aided the communication of resources allocated in each project's activity; and

- The replication of knowledge for the development team helps to mitigate the impact of few experts knowing almost all the solution components, facilitating the resource reallocation.

The specific objectives of the study were: i) to identify the common practices in the field of agile software development and PPM that are adopted by the company; ii) to identify the types of uncertainties found in the portfolio studied; iii) to identify how the company handles these uncertainties; and iv) to make recommendations according to the results of the research and literature review.

### *Common practices in the field of agile software development adopted by the company*

Considering the 12 principles defined in the Manifesto by Beck et al. (2001) and the seven best practices identified in the literature by Leffingwell (2007), it is possible to conclude that the BU tries to deliver early, frequently and continuously working software within a Sprint of two weeks, as recommend by Leffingwell (2007) and Krebs (2009); embraces changing requirements, as stated by one project manager, “changes are in the DNA of the company”, and the development team is aware of the changing environment; the team has engaged people that talk with each other constantly, in different ways according to each project characteristic.

Although the organizational climate is much better now, the development team still lacks the full support of the company, as stated by the project managers. The development of a simple design and complex behavior is a challenge being faced every day by them. Cockburn (2006) said that the architecture needs to be adjusted over the time and grow in steps, but to do it depends on availability and authority, and maturity of the team.

The direct interaction between customers and the development team during the project execution was not identified in the portfolio investigated. The alternative adopted by the BU was the use of intermediaries: business analysts and the project managers. Some contractual terms and the customers’ location contributed to this, but it was not possible to identify other reasons about if it is right or wrong for the context studied.

It was evidenced that there are people working long hours during overtime, affecting the productivity and probably introducing more errors into the code, which reflects, in part, on some reworks identified in the projects. Also, evidences were not found of formal processes to test the software too, but some initiatives are in progress to resolve this fundamental issue, nor of the

development team reflecting on what they do after each regular interval, but just informal conversation during one Daily Scrum in the week.

It is also evidenced the engagement of the development team, especially senior professionals, to improve the continuous process of integration as described by Leffingwell (2007), but in different levels of applicability for each project.

### ***Common practices in the field of PPM adopted by the company***

Although the company has a formal methodology to manage a portfolio of projects, it is not effectively used, and there is not a dedicated division to focus on evolution and innovation of current and new products, which is in part related to concepts studied by Cooper, Edgett and Kleinschmidt (1997a, b), it was possible to identify how the BU handles some processes of PPM suggested by the literature review.

Almost all typical problems that occur in conjunction with inadequate portfolio management identified by Vähäniitty, Rautiainen, and Lassenius (2010) were also found in this case, such as: excessive multitasking, firefighting, overload, and slipping schedules.

Considering the context of the projects, the processes of identification, classification, evaluation, selection, prioritization, authorization, and review, showed to be less relevant compared with the interdependence among projects, and resource constraints shared between projects, two key elements identified in the literature review made by Reyck et al. (2005).

### ***Types of uncertainties found in the portfolio studied***

The section 4.4.1 presented six main types of uncertainties found in the portfolio studied and ten impacts related to its types of uncertainties. Three of six types are related to unknown changes of scope. Petit and Hobbs (2010), reported a similar finding in their results: the main source of uncertainty is related to scope changes.

### ***How the company handles these uncertainties***

The section 4.4.2 presented three actions taken to minimize the impacts of changes in the portfolio studied, despite the perception of the project managers that the BU does not manage uncertainties correctly following common practices in the market. The section 4.4.3 also



presented some initiatives of the development team for reallocation and re-optimization of the resources and capabilities.

## 5.2 CONTRIBUTION

The next sections summarize the contributions for academia and practice.

### 5.2.1 For Academia

This master's research contributes for academia since there are few empirical studies of agile methodologies and the contribution of them on PPM. This exploratory case study showed that Scrum can contribute to PPM, particularly in resource reallocation, and dissemination of knowledge.

### 5.2.2 For Practice

This master's research is contributing to the company studied since it was able to make recommendations according to the results of the research and literature review, as follows:

- To re-implement the formal process to plan and prioritize the project's backlog, instead of keeping it under the responsibility of just one person;
- To make workshops to better explain the concepts of agile methodology and how to take best advantage of it;
- To keep evolving in the use of common practices of agile software development, especially those that there are evidences of lack of application, such as: formal processes of testing and regular reflection; and
- To reevaluate how commercial decision making is affecting the project portfolio, since the aggressive deadline of the projects assumed with the customer is not been met;

### 5.3 LIMITATIONS

Considering the time constraint for the completion of the master research, this study presents the following limitations:

- **The research is based on a single case study:** it is not possible to generalize the findings to all project portfolios, although it helps to understand and make comparisons with similar IT project portfolios.
- **Extension of the literature review:** the literature review was explored within the available time. The uncertainty and dynamic capabilities is a wide field of study that was marginally explored.

### 5.4 FUTURE WORKS

For future works is suggested the following topics:

- Expand the research to a multiple case study of IS providers whose business is based on projects to identify common characteristics in the context of agile software development and PPM;
- Explore how the organizations can better manage constraints related to contractual terms, such as fixed price contract with uncertainty scope, that affect project deadlines and communication between customer and development team;
- Study more in depth the literature of changing management, and the poles presented in this research. Recently, the PMI published two guides that demonstrate the relevance of these subjects: i) *Software Extension to the PMBOK® Guide Fifth Edition*, a standard that provides guidance on the management of software development projects, and bridges the gap between the traditional and iterative approaches, and ii) *Managing Change in Organizations*: a guide that go in-depth on the change management processes. Although they are not academic material, they can contribute with future researches.

## REFERENCES

- Archer, N. P. & Ghasemzadeh, F. (1999). An integrated framework for project portfolio selection. *International Journal of Project Management*, 17(4), 207-216.
- Associação Brasileira das Empresas de Software. (2013). *Brazilian software market: scenario and trends*. Retrieved from <http://central.abessoftware.com.br/Content/UploadedFiles/Arquivos/Dados%202011/publicacao-dados-do-setor-2013.pdf>.
- Beck, M. et al., Beedle, M., Bennekum, A. Van., 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). *The manifesto for agile software development*. Retrieved from <http://www.agilemanifesto.org>
- Brady, T., Davies, A. & Gann, D. M. (2005). Creating value by delivering integrated solutions. *International Journal of Project Management*, 23, 360-365.
- Carvalho, M. M. & Rabechini, R., Jr. (2011). *Fundamentos em gestão de projetos: Construindo competências para gerenciar projetos* (3rd. ed.). São Paulo, SP: Atlas.
- Castro, H. G. & Carvalho, M. M. (2010a). Gerenciamento do portfolio de projetos (PPM): Estudos de caso. *Produção*, 20(3), 303-321.
- Castro, H. G. & Carvalho, M. M. (2010b). Gerenciamento do portfolio de projetos: Um estudo exploratório. *Gest. Prod.*, 17(2), 283-296.
- Cockburn, A. (2006). *Agile software development: The cooperative game* (2nd. ed.). Boston, MA: Addison-Wesley.
- Cooper, R. G., Edgett, S. J. & Kleinschmidt, E. J. (1997a). Portfolio management in new product development: Lessons from the leaders—I. *Research Technology Management*, 40(5), 16-28.

- Cooper, R. G., Edgett, S. J. & Kleinschmidt, E. J. (1997b). Portfolio management in new product development: Lessons from the leaders—II. *Research Technology Management*, 40(6), 43-52.
- Cooper, R. G., Edgett, S. J. & Kleinschmidt, E. J. (1998). Best Practices for managing R&D portfolios. *Research Technology Management*, 41(4), 20-33.
- Dybå, T. & Dingsøy, T. (2008). Empirical studies of agile development: A systematic review. *Information and Software Technology*, 50(9), 833-859.
- Dybå, T. & Dingsøy, T. (2009). What do we know about agile software development? *IEEE Software*, 26(5), 6-9.
- Eisenhardt, K. M. & Martin, J. A. (2000). Dynamic capabilities: What are they? *Strategic Management Journal*, 21(10-11), 1105-1121.
- Fernandez, D. J. & Fernandez, J. D. (2008). Agile Project management: Agilism versus traditional approaches. *The Journal of Computer Information Systems*. 49 (2), 10-17.
- Ghasemzadeh, F. & Archer, N. P. (2000). Project portfolio selection through decision support. *Decision Support Systems*, 29, 73-88.
- Gil, A. C. (2008). *Métodos e técnicas de pesquisa social* (6th. ed.). São Paulo, SP: Editora Atlas.
- Highsmith, J. (2002). *Agile software development ecosystems*. Boston, MA: Addison-Wesley.
- Imbrizi, F. G. & Maccari, E. A. (2013). Gerenciamento de portfólio de projetos de soluções integradas de software: um estudo de caso. *Proceedings of the XVI Simpósio de Administração da Produção, Logística e Operações Internacionais*, São Paulo, 1-15.
- Ionel, N. (2009). Agile software development methodologies: An overview of the current state of research. *The Journal of the Faculty of Economics*, 4(1), 381-385.
- Kerzner, H. (2009). *Project management: A system approach to planning, scheduling, and controlling* (10th. ed.). New York, NY: John Wiley & Sons.

- Krebs, J. (2009). *Agile portfolio management*. Redmond, WA: Microsoft Press.
- Kruchten, P. (2011). Contextualizing agile software development. *Journal of Software: Evolution and Process*, 25(4), 351-361.
- Leffingwell, D. (2007). *Scaling software agility: Best practices for large enterprises*. Upper Saddle River, NJ: Addison Wesley.
- Little, T. (2005). Context-adaptive agility: Managing complexity and uncertainty. *IEEE Software*, 22(3), 28-35.
- Machado, R. F., Sgarbi, E. M., Martins, R. N., Relli, C. S., Reinehr, S. & Malucelli, A. (2011). Gerenciamento de portfólio de projetos: Estado da prática sob a ótica dos gerentes de projetos de grandes organizações de tecnologia da informação. *Revista de Informática Aplicada*, 7(2), 31-40.
- Martins, G. A. & Theóphilo, C. R. (2009). *Metodologia da investigação científica para ciências sociais aplicadas* (2nd. ed.). São Paulo, SP: Atlas.
- Mcfarlan, F. W. (1981). Portfolio approach to information systems. *Harvard Business Review*, 59(5), 142-151.
- Petit, Y. & Hobbs, B. (2010). Project portfolios in dynamic environments: Sources of uncertainty and sensing mechanisms. *Project Management Journal*, 41(4), 46-58.
- Petit, Y. (2011). *Project portfolio in dynamics environments: Organizing for uncertainty*. Ph.D. Dissertation, University of Quebec at Montreal, Montreal, Canada.
- Project Management Institute. (2013a). *A guide to the project management body of knowledge (PMBOK® guide)* (5th. ed). Newtown Square, PA: Project Management Institute.
- Project Management Institute. (2013b). *The standard for portfolio management* (3rd. ed.). Newtown Square, PA: Project Management Institute Inc.
- Project Management Institute. (2013c). *The standard for program management* (3rd. ed.). Newtown Square, PA: Project Management Institute Inc.

- Rautiainen, K., Shantz, J. & Vähäniitty, J. (2011, January). Supporting scaling agile with portfolio management: Case paf.com. *Proceedings of the 44<sup>th</sup> Hawaii International Conference on System Sciences*, Kanui, 1-10.
- Reyck, B. D., Grushka-Cockayne, Y., Lockett, M., Calderini, S. R., Moura, M. & Sloper, A. (2005). The impact of project management on information technology projects. *International Journal of Project Management*, 23(7), 524-537.
- Santos Filho, E. V. (2012). *Relação entre gestão de portfolio de projetos de software e desenvolvimento ágil: Um caso com o framework scrum no setor público*. Master's Thesis, Universidade Católica de Brasília, Brasília, Brazil.
- Schwaber, K. (2004). *Agile project management with scrum*. Redmond, WA: Microsoft Press.
- Schwaber, K. & Sutherland, J. (2013). *The scrum guide*. Retrieved from [http://www.scrum.org/storage/scrumguides/Scrum\\_Guide.pdf](http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf)
- Teece, D. J. (2007). Explicating dynamic capabilities: the nature and microfoundations (of sustainable) enterprise performance. *Strategic Management Journal*, 28, 1319-1350.
- Vähäniitty, J., Rautiainen, K., & Lassenius, C. (2010). Small software organizations need explicit project portfolio management. *IBM Journal of Research and Development*, 54 (2), 1-12.
- West, D., Grant, T., Gerush, M. & D'Silva, D. (2010). *Agile development: Mainstream adoption has changed agility*. Forrester Research.
- West, D., Gilpin, M., Grant, T. & Anderson, A. (2011). *Water-Scrum-Fall is the reality of agile for most organizations today*. Forrester Research.
- Yin, R. K. (2009). *Case study research: Design and methods* (4th. ed.). Thousand Oaks, CA: Sage Publications.

## APPENDIX A – CASE STUDY PROTOCOL

This case study protocol aims to guide the researcher in carrying out the data collection from a single-case study titled “*Agile software development and project portfolio management in dynamic environments: a case study of an integrated solution provider*”.

The research question is “*How agile software development contributes to project portfolio management in dynamic environments of an integrated solutions provider?*”, the main goal is to understand how agile software development contributes to project portfolio management in dynamic environments of an integrated solutions provider, and the specific objectives are: i) to identify the common practices in the field of agile software development and project portfolio management that are adopted by the company; ii) to identify the types of uncertainties found in the portfolio studied; iii) to identify how the company handles these uncertainties; and iv) to make recommendations according to the results of the research and literature review.

This protocol is divided in two parts: I) identification of interviewees; and II) questions based on propositions. Part I has some questions to identify the interviewees, their position and background experience. The interviewees suggested are i) the development manager; ii) the project managers of each projects portfolio; and iii) at least two key professionals of the development team. Part II contains the question of the case study based on the propositions established according to the literature review. The conceptual framework and some questions of the interview guide proposed by Petit (2011) support extensively, direct and indirectly, the development of these questions.

In part II, a documentary research will be necessary to support the investigation findings’ reports, request for proposals (RFP), contracts, plans, technical and functional specifications, reviews, schedules, e-mails, and other types of documents generated in the projects. The following tables describe the case study protocol questions for each part defined before.

<b>PART I – Identification of interviewees</b>
<b>Objective:</b> to identify the interviewees, their position and background experience.
For each development manager, project manager, and key professional of the software development team:
<ul style="list-style-type: none"> <li>a) What is your name?</li> <li>b) What is your current position?</li> <li>c) What are the main responsibilities and challenges in this position?</li> <li>d) How long have you been in the company and in the current position?</li> <li>e) What are the limitations of authority in this position, in practical terms?</li> <li>f) What are your background experiences?</li> <li>g) How long have you been working with traditional and/or agile projects?</li> </ul>

<b>PART II – Questions based on propositions</b>	
<b>Objective:</b> to examine something within the scope of study guided by the theoretical propositions.	
<b>Agile Methodology</b>	<b>PROPOSITION 1: Processes and tools are important, but talented, motivated, valued, skilled, self-organized people and the interaction between them are far more important, thus process and tools must be adapted to people (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006).</b>
	<ul style="list-style-type: none"> <li>1. How people are organized as a team?</li> <li>2. What is the level of expertise of the team?</li> <li>3. How is the perception of organizational climate by team?</li> <li>4. What are the team’s domain knowledge gaps?</li> </ul>
	<b>PROPOSITION 2: Working software delivered early and frequently allows for continued wins, early feedback from users, rapid response to changing marketing conditions, so that the customer can sense that deliverables are evolving, despite of what was documented (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006, Leffingwell, 2007).</b>
	<ul style="list-style-type: none"> <li>5. What is the frequency of releasing software?</li> <li>6. How is the process of evaluating the working software delivered to the customer?</li> <li>7. Is there any process of versioning, building, and releasing? How does it work?</li> <li>8. How long do the projects take from start to product release?</li> </ul>
	<b>PROPOSITION 3: Just the customer collaboration and close communication with the development team during the time of delivery can provide the customers with their real needs and prevent undesirable outcomes (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006).</b>
	<ul style="list-style-type: none"> <li>9. How is the interaction among the development team, business team, and customer?</li> <li>10. How is the process to communicate changes and request new requirements among the development team, business team, and customer?</li> <li>11. Are the market needs well understood?</li> <li>12. How well is the scope defined?</li> </ul>
	<b>PROPOSITION 4: Planning is useful, important and necessary, but adapting to changes to the plan is more important and useful for the customer's competitive advantage, especially in an environment characterized by change, speed, and turbulence (Beck et al., 2001; Highsmith, 2002; Cockburn, 2006, Leffingwell, 2007).</b>
	<ul style="list-style-type: none"> <li>13. How does the company plan to develop software?</li> <li>14. How often does the company review the plan according to changing market conditions?</li> <li>15. How does the company face changes?</li> </ul>



<b>Project Portfolio Management</b>	<b>PROPOSITION 5: A portfolio is a collection of programs, projects, or operations managed as a group to achieve organizational strategies and objectives (PMI, 2013).</b>
	16. How is described the project portfolio in the organization (description, quantity and types of projects, main objectives, etc.)?
	17. How is described each project in the portfolio (description, main objectives, results expected, etc.)?
	18. What is the degree to which other projects depend on this project?
	19. What is the criticality of each project to the customer business?
	<b>PROPOSITION 6: The project portfolio management is defined as a dynamic decision process where projects are evaluated, selected, prioritized, authorized, assessed, and monitored (Cooper, Edgett &amp; Kleinschmidt, 1997a; Reyck et al., 2005; PMI, 2013b).</b>
	20. What are the processes to evaluate, select, prioritize, authorize, assess, and monitor the project portfolio?
21. Who is involved in this process of decision making?	
22. What are the criteria to prioritize projects?	
<b>Uncertainty and Dynamic Capabilities</b>	<b>PROPOSITION 7: The effective allocation of resources, increasingly scarce, will make all the difference in achieving the objectives set by the organization (Cooper, Edgett &amp; Kleinschmidt, 1997a; Reyck et al., 2005; PMI, 2013b).</b>
	23. What are the criteria to plan and to allocate human resources to the projects?
	24. Is the planning of resource allocation done either individually or collectively, considering the ongoing projects and proposals for new projects?
	25. Is the reallocation of human resources projects communicated to all stakeholders?
	26. What is the team size (the median during the project execution)?
	27. How the team is distributed geographically?
	<b>PROPOSITION 8: Portfolio management might implement process to manage and control uncertainty, and not only monitor changes (Petit &amp; Hobbs, 2010; Petit, 2011).</b>
28. How does the company manage and control uncertainty?	
29. How does the company handle changes? Are there any mechanisms to facilitate embracing changes?	
<b>Uncertainty and Dynamic Capabilities</b>	<b>PROPOSITION 9: There are additional types and sources of change that the organizations managing the project portfolio were facing, beyond the commonly described in the literature (Petit &amp; Hobbs, 2010).</b>
	30. What are the sources of uncertainty identified in the portfolio?
	31. What types of changes are identified during the software development, project and portfolio management? What are the impacts of these changes to the projects and portfolio?
	32. What is the frequency of changes in the project and portfolio? How many changes were required in period investigated?
	<b>PROPOSITION 10: It is necessary to reallocate and re-optimize resources and capabilities to adapt to changing environments (Petit &amp; Hobbs, 2010; Petit, 2011).</b>
33. How human resources and capabilities are addressed to handle the uncertainties?	
34. Is there any specific training to prepare the human resources to deal with this changing environment? How is it applied?	
35. Are there new domain technologies added to an existing product?	

## APPENDIX B - DEVELOPMENT TEAM MEMBER OF THE PRODUCT 1

This table shows an overview of the BU's professionals under the development manager of the product 1.

Professional	Experience Level	Graduation Year	Trainee Experience in IT (years)	Professional Experience in IT (years)*
Development Coordinator	Senior	2003	3	13
Developer 01 / Chief Architect	Senior	1999	0.8	15.2
Developer 02 / Web Architect	Senior	2004	2	14
Developer 03	Senior	2004	2	8
Developer 04	Senior	1999	2	12
Developer 05	Senior	1994	0.2	21
Developer 06	Senior	2005	2	11
Developer 07	Senior	2000	1	12
Developer 08	Intermediate	2009	4	4.5
Developer 09	Intermediate	2009	1	3.5
Developer 10	Junior	2011	1	2
Developer 11	Junior	2010	1	1.5
Developer 12	Junior	2011	2	2
Developer 13	Junior	2013	0.7	1.1
Developer 14	Junior	2013	3	0.5
Developer 15	Junior	2013	1	0.5
Developer 16	Trainee	2015	0.3	0
Developer 17	Trainee	2014	1	0
Developer 18	Trainee	2015	0.3	0
Developer 19	Trainee	2014	0.4	0
Developer 20	Trainee	2014	2.3	0
Developer 21	Trainee	2014	0.7	0
Developer 22	Trainee	2014	0.7	0
DBA 01	Senior	1997	0.5	16.5
DBA 02	Senior	1994	0	12
DBA 03	Trainee	2015	2	0
Tester 01	Junior	2010	2	3.3
Tester 02	Junior	2011	1	2
Tester 03	Trainee	2015	2	0
Tester 04	Trainee	2016	0.5	0
Tester 05	Trainee	2016	1	0
Tester 06	Trainee	2016	0.2	0

\* Excluding experience during traineeship

**APPENDIX C – SPRINT REPORTING TEMPLATE**

**Sprint:** 47-48/13      **Period:** 11/18/2013 to 11/29/2013

Issues	Estimated Resource	Status	NS	P	R	Po
<b>Project Alpha</b>						
<b>Issues not done from previous planned Sprint</b>						
Issue 01 – description of the issue 01	Developer 01	Not started	2	X		
Issue 03 – description of the issue 02	Developer 02; DBA 01	In progress	3			
<b>New issues included in the middle of the Sprint</b>						
Issue 04 – description of the issue 04	Developer 04	Not started	1			
Issue 05 – description of the issue 05	DBA 02	Done	1	X		
<b>Planned issues</b>						
Issue 07 – description of the issue 07	Developer 06	Not started	1			
Issue 08 – description of the issue 08	DBA 02	Done	1			

NS: Number of the Sprint / P: Issue in production / R: Recurrent issue / Po: Issue postponed

<b>SPRINT 47-48/13 – Planned and New Issues</b>																			
Project	Not Started			In Progress			Test			Done			Canceled			Total			
	A	P	N	A	P	N	A	P	N	A	P	N	A	P	N	A	P	N	Total
Alpha	1	0	0	2	0	2	0	0	1	2	0	6	0	0	0	5	0	9	14
Beta	1	0	0	0	1	0	2	0	0	1	3	5	0	0	0	4	4	5	13
Gama	2	2	0	6	2	0	0	0	0	27	2	8	0	0	0	35	6	8	49
Total	4	2	0	8	3	2	2	0	1	30	5	19	0	0	0	44	10	22	76
	5.3%	2.6%	0.0%	10.5%	3.9%	2.6%	2.6%	0.0%	1.3%	39.5%	6.6%	25.0%	0.0%	0.0%	0.0%	57.9%	13.2%	28.9%	
	7.9%			17.1%			3.9%			71.1%			0.0%			100.0%			

A: Issues not done from previous planned Sprint / P: Planned issued / N: New issues included in the middle of the Sprint