



Daniela Milagros Quenaya Mendoza

**Gerenciamento e Planejamento de Projetos de Software
Usando Metodologias Ágeis: Um Estudo de Caso**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia de Produção do Departamento de Engenharia Industrial da PUC-Rio.

Orientador: Prof. Nélio Domingues Pizzolato

Rio de Janeiro

Abril de 2011



Daniela Milagros Quenaya Mendoza

Gerenciamento e Planejamento de Projetos de Software
Usando Metodologias Ágeis: Um Estudo de Caso

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia de Produção da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Nélio Domingues Pizzolato
Orientador
DEI PUC-Rio

Prof. Luiz Felipe Roris Rodriguez Scavarda do Carmo
DEI PUC-Rio

Prof. Eugenio Kahn Epprecht
DEI PUC-Rio

Prof. José Eugenio Leal
Coordenador(a) Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 11 de Abril de 2011

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor do orientador.

Daniela Milagros Quenaya Mendoza

Graduou-se na Faculdade de Engenharia Industrial da Universidade Nacional de San Agustín de Arequipa (Peru) em 2007. Trabalhou em diversas empresas no Peru principalmente nas áreas de gerenciamento de projetos, gestão de custos e planejamento estratégico.

Ficha Catalográfica

Quenaya Mendoza, Daniela Milagros

Gerenciamento e planejamento de projetos de software usando metodologias ágeis : uma estudo de caso / Daniela Milagros Quenaya Mendoza ; orientador: Nélio Domingos Pizzolato. – 2011.

104 f. : il. (color.) ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, 2011.

Inclui bibliografia

1. Engenharia Industrial – Teses. 2. Gerenciamento de projetos. 3. Gerenciamento de projetos de software. 4. Metodologias ágeis. 5. Metodologias tradicionais. 6. Produção enxuta de software. 7. Qualidade de software. I. Pizzolato, Nélio Domingues. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Industrial. III. Título.

CDD: 658.5

A minha família, pelo apoio ao longo do período
da elaboração deste trabalho.

Agradecimentos

Ao Departamento de Engenharia Industrial da PUC-Rio pela oportunidade de ser parte dele, fazendo o curso de Mestrado.

Ao Professor Nélío Pizzolato pela sua colaboração e apoio durante a elaboração deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela ajuda financeira.

A Javier pela amizade e apoio para desenvolver esta pesquisa.

A Edison pela oportunidade de testar este estudo na sua organização.

Aos Professores Epprecht e Luiz Felipe pelas correções finais na escrita do texto.

Finalmente agradecimentos especiais a Berioska, Guina, Orivalde Soares, Juan Carlos pela motivação, amizade e apoio nesta etapa do mestrado.

Resumo

Quenaya Mendoza, Daniela Milagros; Pizzolato, Nélío. **Gerenciamento de projetos de software usando metodologias ágeis: Um Estudo de Caso.** DEI PUC-Rio, 2011. 104 p. Dissertação de Mestrado - Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Hoje em dia, entregar *software* com qualidade, nos prazos previstos e respeitando os custos é um grande diferencial para empresas dedicadas a esta atividade. A constante procura de produtividade e qualidade em projetos de *software* originou diversos modelos de gerenciamento que procuram auxiliar esta área com metodologias e formas de organização de trabalho. O modelo mais conhecido na área de gerenciamento de *software* é o “Modelo em Cascata”, caracterizado pela rigidez e fases bem definidas nos processos. Uma vez encontrados problemas com os modelos rígidos surgiram como alternativa de gerenciamento os métodos ágeis, caracterizados pela agilidade, iteratividade e adaptabilidade para diversos projetos. Diante isto o presente trabalho analisa o problema do gerenciamento de projetos de desenvolvimento de *software*, e propõe um modelo de referencia de gerenciamento de projetos de *software* considerando a abordagem tradicional de gerenciamento de projetos assim como a abordagem ágil. A pesquisa analisa a compatibilidade das metodologias ágeis com o *Capability Maturity Model Integration* (CMMI) que é um modelo de maturidade nos processos de desenvolvimento de *software*. Finalmente a pesquisa apresenta um estudo de caso, que permitiu observar as forças e fraquezas dos métodos estudados, concluindo que as praticas ágeis contribuem na melhoria e gerenciamento de projetos de *software*.

Palavras-chave

Gerenciamento de Projetos; Gerenciamento de Projetos de Software; Metodologias Ágeis; Metodologias Tradicionais; Produção Enxuta de Software; Qualidade de Software.

Abstract

Quenaya Mendoza, Daniela Milagros; Pizzolato, Nélío. **Managing Software Projects Using Agile Methodologies: A Case Study**. DEI PUC-Rio, 2011. 104 p. MSc. Dissertation - Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Today, delivering quality software, on time and respecting costs is a key differentiator for companies dedicated to this activity. The constant search for productivity and quality in software project management led to several models that seek to help this area with methods and forms of work organization. The best known model in the area of software management is the "Waterfall Model", characterized by stiffness and well-defined phases in the process. Once found problems with rigid models have emerged as alternative management of agile methods, characterized by agility, and adaptability to various iterative projects. Given that this paper analyzes the problem of managing software development projects, and proposes a reference model for managing software projects considering the traditional approach of project management as well as the agile approach. The research examines the compatibility of agile methodologies with the Capability Maturity Model Integration (CMMI) is a maturity model in software development processes. Finally the research presents a case study, which allowed us to observe the strengths and weaknesses of the methods studied, concluding that such practices contribute to improved and agile project management software.

Keywords

Project management; Software project management; Agile methodologies, Traditional methodologies; Lean production of software; Software quality.

Sumário

1 Introdução	13
1.1. Preliminares	13
1.2. Justificativa do Estudo	14
1.3. Objetivos da Pesquisa	14
1.4. Delimitação do Estudo	15
1.5. Estrutura da Dissertação	16
2 Revisão Bibliográfica	17
2.1. Considerações Iniciais	17
2.2. Contextualização do Problema	17
2.3. Metodologias Tradicionais de Desenvolvimento de Software	21
2.4. Metodologias Ágeis de Desenvolvimento de <i>Software</i>	23
2.4.1. Extreme Programming	25
2.4.2. SCRUM	26
2.4.3. Feature Drive Development	29
2.5. A Produção Enxuta de <i>Software</i>	30
2.6. Projetos	33
2.6.1. Gerenciamento de Projetos	34
2.6.2. Processos de Gerenciamento de Projetos	35
2.6.3. Ciclo de Vida do Projeto	36
2.7. Gerenciamento da Qualidade do Projeto	38
2.7.1. Processos de Gerenciamento da Qualidade	38
2.7.2. Metas da Qualidade e Ferramentas de Qualidade	39
2.7.3. Gerência da Qualidade em <i>Software</i>	41
2.7.4. CMMI	42
2.8. Diferenças entre as Metodologias ágeis e tradicionais	47
2.9. Combinando Scrum com XP	49
2.10. Características Comuns Scrum e CMMI	50
2.11. Considerações Finais	52

3 Formulação da Metodologia	53
3.1. Considerações Iniciais	53
3.2. Proposta de Modelo de Gerenciamento de Projetos de <i>Software</i>	54
3.2.1. Início	56
3.2.2. Planejamento Inicial do Projeto	59
3.2.3. Planejamento Detalhado ou Planejamento das Iterações	60
3.2.4. Execução do Projeto	63
3.2.5. Encerramento do Projeto	65
3.2.6. Monitoramento e Controle do Projeto	67
4 Estudo de Caso	69
4.1. Considerações iniciais	69
4.2. Descrição do Processo de Desenvolvimento e Tipos de Projetos/ Soluções da Empresa	69
4.2.1. Soluções sob encomenda	70
4.2.2. Soluções pré-formatadas	70
4.2.3. Soluções embarcadas	70
4.3. Aplicação da Metodologia no Caso de Estudo Proposto	70
4.3.1. Início	71
4.3.2. Planejamento Inicial do Projeto	74
4.3.3. Planejamento Detalhado	75
4.3.4. Execução do Projeto	79
4.3.5. Monitoramento e Controle das atividades de trabalho	81
4.3.6. Encerramento do Projeto	88
5 Considerações Finais	90
6 Referências Bibliográficas	92
7 Apêndice I : Princípios XP	95
Apêndice II: Responsabilidades XP	97
Apêndice III: Etapas XP	98
Apêndice IV: Níveis de Maturidade CMM	100

Apêndice V: Fator de Ajuste da Equipe de Trabalho Testada	102
Apêndice VI: Características comuns das metodologias usadas no	
Modelo de Referencia	104

Lista de figuras

Figura 2.1 – Desenvolvimento tradicional (em cascata)	22
Figura 2.2 – Custo de mudança do <i>Software</i>	25
Figura 2.3 – Desenvolvimento de <i>Software</i> com XP	25
Figura 2.5 – Fases Scrum	27
Figura 2.7 – : Quadro de cartões de funcionalidades	32
Figura 2.8 Grupos de processo de gerenciamento de projetos.	35
Figura 2.9 – Nível de custos e pessoal no ciclo de vida do projeto	37
Figura 2.10 – Influência das partes interessadas ao longo do tempo	37
Figura 2.11 – Processos de Gerenciamento da Qualidade	39
Figura 2.12 – Níveis de Maturidade do CMM , Fiorini 1998	43
Figura 3.1 –Macro e Micro Componentes do Modelo Proposto	54
Figura 3.3 – Iniciação: Autorização do projeto a desenvolver	56
Figura 3.5 – Arquitetura Inicial do Projeto	59
Figura 3.6 – Planejamento Inicial	60
Figura 3.7 – Importância dos cartões de funcionalidades	61
Figura 3.8 – Cálculo de Velocidade das Iterações “SCRUM”	62
Figura 3.9 – Planejamento Detalhado	63
Figura 3.11 – Encerramento do Projeto	66
Figura 3.12 – Gráficos de monitoramento e controle	67
Figura 3.13 – Gráficos de monitoramento e controle	68
Figura 4.1 – Inicio	71
Figura 4.2 – Planejamento Inicial	74
Figura 4.3 – Planejamento Detalhado	75
Figura 4.4 – Planejamento Inicial	79
Figura 4.5 – Cenário de Uso das Iterações do Projeto	80
Figura 4.6 – Atividades de Trabalho da Equipe	83
Figura 4.8 – Previsto vs Realizado no Projeto Riscos	85
Figura 4.9 – Previsto vs Realizado nas Tarefas do Projeto Riscos	85

Lista de tabelas

Tabela 2.1: Distribuição por Setor da Economia	18
Tabela 2.3: Fatores de Sucesso em Projetos de <i>Software</i>	20
Tabela 2.4: Fatores de Fracasso em Projetos <i>Software</i>	20
Tabela 2.5 – Responsabilidades Scrum	28
Tabela 2.6 – Elementos no Scrum	29
Tabela 2.7 – Fatores de Qualidade para as Metodologias Ágeis	42
Tabela 2.10 – Categorias da Representação Estagiada CMMI	46
Tabela 2.11 – Diferencias entre as metodologias tradicionais e ágeis	48
Tabela 2.12 – Características comuns entre XP e Scrum	49
Tabela 2.13 – Características comuns entre Scrum e CMMI	51
Tabela 4.1 – Listado da Carteira de Projetos	72
Tabela 4.2 – Backlog do Produto: Visão Inicial dos Processos do Projeto Smart Riscos	73
Tabela 4.3 – Descrição Genérica do Projeto Riscos	73
Tabela 4.4 – Planificação Geral Riscos	75
Tabela 4.5 – Planificação Detalhada da Equipe	76
Tabela 4.6 – Planificação Detalhada da Coordenador Riscos	78
Tabela 4.7 – Planilha Backlog	82
Tabela 7.1 – Princípios XP	95
Tabela 7.2 – Responsabilidades XP: A equipe do cliente	97
FONTE: Adaptado de Extreme Programming Guia Pratica (2002)	97
Tabela 7.3 – Responsabilidades XP: A equipe do desenvolvimento	97
Tabela 7.4 – Etapas XP: Conceitualização do sistema	98
FONTE: Adaptado de Extreme Programming Guia Pratica (2002)	98
FONTE: Adaptado de Extreme Programming Guia Pratica (2002)	99
Tabela 7.6 – Etapas XP: Desenvolvimento	99
Tabela 7.7 – Níveis de Maturidade CMM	100