



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE QUIXADÁ**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**MICAELLY PRISCILA SOARES E SILVA**

**DEFINIÇÃO E IMPLANTAÇÃO DE UM PROCESSO DE SOFTWARE PARA O  
NÚCLEO DE PRÁTICAS DE UMA UNIVERSIDADE**

**QUIXADÁ**

**2013**



**MICAELLY PRISCILA SOARES E SILVA**

**DEFINIÇÃO E IMPLANTAÇÃO DE UM PROCESSO DE SOFTWARE PARA O  
NÚCLEO DE PRÁTICAS DE UMA UNIVERSIDADE**

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação.

Orientadora: Profa. Msc. Carla Ilane Moreira Bezerra

**QUIXADÁ**

**2013**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

- 
- S581d Silva, Micaelly Priscila Soares e  
Definição e implantação de um processo de software para o Núcleo de Práticas de uma universidade / Micaelly Priscila Soares e Silva. – 2013.  
178 f. : il. color., enc. ; 30 cm.
- Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2013.  
Orientação: Profa. MSc. Carla Ilane Moreira Bezerra  
Área de concentração: Computação
1. Software - Desenvolvimento 2. Desenvolvimento ágil de software 3. Modelos de capacitação e maturidade - Software I. Título.

**MICAELLY PRISCILA SOARES E SILVA**

**DEFINIÇÃO E IMPLANTAÇÃO DE UM PROCESSO DE SOFTWARE PARA O  
NÚCLEO DE PRÁTICAS DE UMA UNIVERSIDADE**

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação.

Orientadora: Profa. Msc. Carla Ilane Moreira Bezerra

Aprovada em 17/07/2013.

**BANCA EXAMINADORA**

---

Profa. Msc. Carla Ilane Moreira Bezerra (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Msc. Camilo Camilo Almendra  
Universidade Federal do Ceará (UFC)

---

Prof. Msc. Enyo José Tavares Gonçalves  
Universidade Federal do Ceará (UFC)



Dedico este trabalho à minha família e amigos.



## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, sem ele nada disso seria possível.

Agradeço a meus pais Maristela e Edvaldo, por terem acreditado e investido em mim e por terem dividido essa jornada comigo.

Agradeço a minha família, por estar sempre presente, mesmo que distante, sendo a base de tudo que sou hoje.

Agradeço a professora Carla Ilane, minha orientadora, pela sua disponibilidade constante em revisar, sugerir e auxiliar no melhor andamento deste trabalho. Pela sua orientação exemplar, que sem dúvida é uma das melhores desta universidade. Pelas orientações acadêmica e profissional que me proporcionou por todo esse período.

Agradeço aos professores Enyo José e Camilo Almendra que muito contribuíram para a construção e melhoramento deste trabalho.

Agradeço aos colegas e amigos que direta ou indiretamente me ajudaram, dando-me seu apoio e motivação para a continuação do trabalho.



## RESUMO

Processos de software são fundamentais nas organizações que visam padronizar o seu desenvolvimento de software e obter maior qualidade no produto final. Por isso, muitas organizações estão em busca de definir seus processos. Porém, essa não é uma tarefa simples. A definição e implantação de um processo de software precisa ser bem planejada e executada para que se consiga atingir seus objetivos. Diversas empresas já trabalham com processo de software e algumas universidades estão começando a perceber a sua importância. Dentro de algumas universidades existem fábricas de software que desenvolvem softwares, seja para pesquisa, para suprir a demanda interna ou externa. Algumas dessas fábricas têm como objetivo acolher alunos que estejam cursando a disciplina de estágio e esse é o caso do Núcleo de Práticas de Informática (NPI) da UFC – Campus Quixadá. Para apoiar o desenvolvimento de software, elas estão buscando definir seus processos. Alguns trabalhos de melhoria de processos de software foram realizados no âmbito acadêmico, porém apesar dos esforços, não foi localizado na literatura, trabalhos aplicados em uma fábrica de software semelhante ao NPI. Nesse contexto, esse trabalho apresenta a definição e implantação de um processo de software no Núcleo de Práticas de Informática da Universidade Federal do Ceará. O processo foi construído baseado em boas práticas que foram selecionadas de modelos de processos de software tradicionais e ágeis como CMMI, MPS.BR, RUP, PMBoK, Scrum e SCORE, e na análise das atividades desenvolvidas no NPI, focando nas disciplinas de Gerência de Projetos, Requisitos e Gerência de Configuração. O processo foi implantado em projetos pilotos do NPI. Esses foram acompanhados e os seus resultados foram coletados e analisados. A partir da análise, foram percebidas algumas dificuldades, lições aprendidas, oportunidades de melhoria, bem como as contribuições e limitações deste trabalho.

**Palavras-chave:** Processos de software, Modelos de Qualidade, Melhoria de Processo.



## ABSTRACT

Software processes are essential in organizations which aim to standardize their software development process and assure quality in their final product. Therefore, many organizations seek to define their processes. However, this is not an easy task. The definition and deployment of a software process has to be well planned and executed in order to achieve its goals. Several companies have already worked with software process and some universities are beginning to realize its importance. In some universities, there are software factories that develop software products, for research, or to fulfill demand from their own departments or outside requests from the community. Some of these factories aim to receive students who are enrolled in the internship course, that is the case of the Informatics Practice Center (NPI) at UFC in Quixadá. To aid the software development process, they are trying to define their processes. Some works on process improvement were done in an academic scope, however despite to efforts, was not found in the literature, applied work in a software factory similar to the NPI. In this context, this work presents the definition and implantation of a software process in the Informatics Practice Center (NPI) in the Federal University of Ceará. The process was built based on good practices selected from traditional and software models such as CMMI, MPS.BR, RIP, PMBoK, Scrum and SCORE, in the analysis of the activities developed in the NPI, focusing in Project Management, Requirements and Configuration Management disciplines. The process was deployed in pilot projects from the NPI. These projects had their progress tracked and their results were collected and analyzed. From the analysis, difficulties, lessons learned, improvement opportunities, as well as contributions and limitations on this work were discussed.

**Keywords:** Software Processes, Quality Improvement, Process Improvement



## LISTA DE QUADROS

Quadro 1 – Áreas de processo – CMMI por estágios.....	40
Quadro 2 – Níveis de maturidade do MR-MPS .....	42
Quadro 3 – Resumo das características da metodologia ágil Scrum.....	48
Quadro 4 – Fases da implantação do processo do SERPRO.....	52
Quadro 5 – Fases do processo baseado na abordagem IDEAL.....	54
Quadro 6 – Benefícios obtidos com a implantação do processo do CCA SJ .....	55
Quadro 7 – Benefícios obtidos com a implantação do processo do LENS .....	56
Quadro 8 – Fases do processo do LUPA.....	58
Quadro 9 – Perguntas e respostas da entrevista.....	62
Quadro 10 – Mapeamento das boas práticas dos modelos e o processo do NPI.....	64
Quadro 11 - Mapeamento do artefatos dos modelos e o processo do NPI.....	65
Quadro 12 – Representação da tarefa “Prospectar Projetos” .....	69
Quadro 13 – Representação da tarefa “Realizar Reunião de <i>Kickoff</i> ” .....	69
Quadro 14 – Representação da tarefa “Elaborar Relatório Inicial” .....	70
Quadro 15 – Representação da tarefa “Elicitar Requisitos”.....	71
Quadro 16 – Representação da tarefa “Validar Requisitos”.....	72
Quadro 17 – Representação da tarefa “Criar <i>Backlog</i> do Produto” .....	73
Quadro 18 – Representação da tarefa “Realizar Reunião de Planejamento da <i>Sprint</i> ” .....	75
Quadro 19 – Representação da tarefa “Especificar Caso de Uso” .....	77
Quadro 20 – Representação da tarefa “Preparar Ambiente”.....	79
Quadro 21 – Representação da tarefa “Implementar Caso de Uso”.....	80
Quadro 22 – Representação da tarefa “Gerar Release”.....	80
Quadro 23 – Representação da tarefa “Especificar Caso de Teste” .....	81
Quadro 24 – Representação da tarefa “Executar Teste”.....	82
Quadro 25 – Representação da tarefa “Reportar Defeitos”.....	83
Quadro 26 – Representação da tarefa “Realizar Reunião de Revisão da <i>Sprint</i> ”.....	84
Quadro 27 – Representação da tarefa “Realizar Reunião de Status do NPI” .....	86
Quadro 28 – Representação da tarefa “Realizar Reunião Diária”.....	86
Quadro 29 – Representação da tarefa “Criar Plano de Gerenciamento de Configuração” .....	88
Quadro 30 – Representação da tarefa “Estabelecer um Sistema de Gestão de Configuração”.....	89
Quadro 31 – Representação da tarefa “Elaborar Relatório Final”.....	90
Quadro 32 – Representação da tarefa “Realizar Reunião de Encerramento do Projeto” .....	91



Quadro 33 – <i>Checklist</i> de aderência do processo do NPI – Atividade Iniciar Projeto .....	94
Quadro 34 – <i>Checklist</i> de aderência do processo do NPI – Atividade Requisitos .....	96
Quadro 35 – <i>Checklist</i> de aderência do processo do NPI – Atividade Gerenciamento do Projeto.....	98
Quadro 36 – <i>Checklist</i> de aderência do processo do NPI – Atividade Gerenciamento de Configuração .....	99



## LISTA DE FIGURAS

Figura 1 - Arquitetura Geral do RUP .....	46
Figura 2 – Fluxo principal do processo de software do Núcleo de Práticas de Informática ....	67
Figura 3 – Fluxo da atividade “Iniciar Projeto” .....	68
Figura 4 – Fluxo da atividade “Requisitos” .....	71
Figura 5 – Fluxo da iteração “ <i>Sprint</i> ” .....	74
Figura 6 – Fluxo da atividade “Planejar <i>Sprint</i> ” .....	75
Figura 7 – Fluxo da atividade “Especificar Caso de Uso” .....	76
Figura 8 – Fluxo da atividade “Implementação” .....	79
Figura 9 – Fluxo da atividade “Testes” .....	81
Figura 10 – Fluxo da atividade “Revisão da <i>Sprint</i> ” .....	83
Figura 11 – Fluxo da atividade “Gerenciamento do Projeto” .....	85
Figura 12 – Fluxo da atividade “Gerenciamento de Configuração” .....	87
Figura 13 – Fluxo da atividade “Encerrar Projeto” .....	90



## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AMP	Avaliação e Melhoria do Processo Organizacional
AQU	Aquisição
CAR	Análise e Resolução de Causas
CCA SJ	Centro de Computação da Aeronáutica de São José dos Campos
CERCOMP	Centro de Recursos Computacionais
CM	Gerência de Configuração
CMM	Modelo de Maturidade e de Capacidade
CMMI	Modelo Integrado de Maturidade e de Capacidade
CMMI-DEV	Modelo Integrado de Maturidade e de Capacidade para Desenvolvimento
COPPE	Coordenação de Programas de Pós-Graduação em Engenharia
CRUD	<i>Create, Read, Update e Delete</i>
DAR	Análise e Tomada de Decisões
DFP	Definição do Processo Organizacional
DRE	Desenvolvimento de Requisitos
DRU	Desenvolvimento para Reutilização
ES	Engenharia de Software
EPF	<i>Eclipse Process Framework</i>
EUA	Estados Unidos da América
GC	Gestão de Configuração
GCO	Gerência de Configuração
GDE	Gerência de Decisões
GPP	Gerência de Portfólio de Projetos
GPR	Gerência de Projetos
GQA	Garantia da Qualidade
GRE	Gerência de Requisitos
GRI	Gerência de Riscos
GRH	Gerência de Recursos Humanos
GRU	Gerência de Reutilização
IBM	<i>International Business Machines</i>
IDE	<i>Integrated Development Environment</i>



IDEAL	<i>Integrating, Diagnosing, Establishing, Acting &amp; Learning</i>
IEC	<i>International Electrotechnical Commission</i>
IPM	Gestão Integrada de Projeto
ISO	<i>International Organization for Standardization</i>
ITP	Integração do Produto
LENS	Laboratório de Engenharia de Software
LUPA	Laboratório de Aplicações Ubíquas e Pervasivas
MA	Medição e Análise
MA-MPS	Método de Avaliação
MED	Medição
MN-MPS	Modelo de Negócio
MPS.BR	Melhoria de Processo de Software Brasileiro
MR-MPS	Modelo de Referência
NPI	Núcleo de Práticas de Informática
OPD	Definição dos Processos da Organização
OPF	Foco nos Processos da Organização
OPM	Gestão do Desempenho da Organização
OPP	Desempenho dos Processos da Organização
OT	Treinamento da Organização
P-CMM	<i>People – Capability Maturity Model</i>
PCP	Projeto e Construção do Produto
PI	Integração de Produto
PMBok	Guia do Conhecimento em Gerenciamento de Projetos
PMC	Monitoramento e Controle do Projeto
PMI	<i>Project Management Institute</i>
PP	Planejamento de Projeto
PPQA	Garantia da Qualidade de Processo e Produto
QPM	Gestão Quantitativa de Projeto
RD	Desenvolvimento de Requisitos
REQM	Gestão de Requisitos
RSKM	Gestão de Riscos
RUP	Processo Unificado da Rational
SA-CMM	<i>Software Acquisition – Capability Maturity Model</i>
SAM	Gestão de Contrato com Fornecedores



SCORE	<i>Scrum for Research</i>
SE-CMM	<i>Systems Engineering – Capability Maturity Model</i>
SEI	<i>Software Engineering Institute</i>
SERPRO	Serviço Federal de Processamento de Dados
SI	Sistemas de Informação
SPI-KM	<i>Software Process Improvement Strategy Supported by Knowledge Management</i>
SOFTEX	Associação para a Promoção da Excelência do Software Brasileiro
SVN	<i>Subversion</i>
SW-CMM	Modelo de Maturidade e de Capacidade para Software
TS	Solução Técnica
UFC	Universidade Federal do Ceará
UFG	Universidade Federal de Goiás
UFRJ	Universidade Federal do Rio de Janeiro
VER	Verificação
VAL	Validação
XP	<i>Extreme Programming</i>



## SUMÁRIO

1	INTRODUÇÃO.....	31
1.1	OBJETIVOS.....	33
1.2	ESTRUTURA DO TRABALHO .....	33
2	FUNDAMENTAÇÃO TEÓRICA .....	35
2.1	PROCESSO DE DESENVOLVIMENTO DE SOFTWARE .....	35
<b>2.1.1</b>	<b>Definição de Processo de Software .....</b>	<b>36</b>
2.2	MODELOS DE MELHORIA DE PROCESSOS DE SOFTWARE .....	37
<b>2.2.1</b>	<b>Modelo Integrado de Maturidade e de Capacidade (CMMI).....</b>	<b>38</b>
<b>2.2.2</b>	<b>Modelo de Melhoria de Processo de Software Brasileiro (MPS.BR) .....</b>	<b>40</b>
2.3	MODELO DE GERENCIAMENTO DE PROJETOS .....	41
<b>2.3.1</b>	<b>Guia do Conhecimento em Gerenciamento de Projetos (PMBok) .....</b>	<b>42</b>
2.4	METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE .....	44
<b>2.4.1</b>	<b>Processo Unificado da Rational (RUP).....</b>	<b>45</b>
<b>2.4.2</b>	<b><i>Scrum</i> .....</b>	<b>47</b>
<b>2.4.3</b>	<b><i>Scrum for Research (SCORE)</i> .....</b>	<b>50</b>
2.5	DEFINIÇÃO E IMPLANTAÇÃO DE PROCESSOS DE SOFTWARE .....	50
<b>2.5.1</b>	<b>Processos de Software nas Empresas .....</b>	<b>51</b>
2.5.1.1	Serviço Federal de Processamento de Dados (SERPRO) .....	51
2.5.1.2	Rightway Consultoria & Sistemas .....	51
2.5.1.3	Grupo de empresas .....	53
2.5.1.4	Empresa de Médio Porte .....	53
<b>2.5.2</b>	<b>Processos de Software no Ambiente Acadêmico .....</b>	<b>54</b>
2.5.2.1	Centro de Computação da Aeronáutica.....	54
2.5.2.2	Laboratório de Engenharia de Software da COPPE/UFRJ .....	55
2.5.2.3	Centro de Recursos Computacionais da UFG.....	56
2.5.2.4	Laboratory for Ubiquitous and Pervasive Applications (LUPA) da UFG .....	57
2.6	CONCLUSÃO DA SEÇÃO .....	58
3	PROCEDIMENTOS METODOLÓGICOS .....	59
3.1	CONCLUSÃO DA SEÇÃO .....	60



4	DEFINIÇÃO DO PROCESSO DO NPI .....	61
4.1	PRIMEIRA ETAPA .....	61
4.2	SEGUNDA ETAPA .....	63
4.3	TERCEIRA ETAPA .....	65
4.4	QUARTA ETAPA .....	65
4.5	CONCLUSÃO DA SEÇÃO .....	66
5	PROCESSO DO NPI.....	67
5.1	ATIVIDADE “INICIAR PROJETO” .....	68
<b>5.1.1</b>	<b>Tarefa “Prospectar Projetos” .....</b>	<b>68</b>
<b>5.1.2</b>	<b>Tarefa “Realizar Reunião de Kickoff” .....</b>	<b>69</b>
<b>5.1.3</b>	<b>Tarefa “Elaborar Relatório Inicial” .....</b>	<b>70</b>
5.2	ATIVIDADE “REQUISITOS” .....	70
<b>5.2.1</b>	<b>Tarefa “Elicitar Requisitos” .....</b>	<b>71</b>
<b>5.2.2</b>	<b>Tarefa “Validar Requisitos” .....</b>	<b>72</b>
<b>5.2.3</b>	<b>Tarefa “Criar Backlog do Produto” .....</b>	<b>73</b>
5.3	ITERAÇÃO “SPRINT” .....	73
<b>5.3.1</b>	<b>Atividade “Planejar Sprint” .....</b>	<b>74</b>
5.3.1.1	Tarefa “Realizar Reunião de Planejamento da Sprint” .....	75
<b>5.3.2</b>	<b>Atividade “Especificar Caso de Uso” .....</b>	<b>76</b>
5.3.2.1	Tarefa “Especificar Caso de Uso” .....	76
<b>5.3.3</b>	<b>Atividade “Implementação” .....</b>	<b>78</b>
5.3.3.1	Tarefa “Preparar Ambiente” .....	79
5.3.3.2	Tarefa “Implementar Caso de Uso” .....	80
5.3.3.3	Tarefa “Gerar Release” .....	80
<b>5.3.4</b>	<b>Atividade “Testes” .....</b>	<b>81</b>
5.3.4.1	Tarefa “Especificar Caso de Teste” .....	81
5.3.4.2	Tarefa “Executar Teste” .....	82
5.3.4.3	Tarefa “Reportar Defeitos” .....	82
<b>5.3.5</b>	<b>Atividade “Revisão da Sprint” .....</b>	<b>83</b>
5.3.5.1	Tarefa “Realizar Reunião de Revisão da Sprint” .....	84
5.4	ATIVIDADE “GERENCIAMENTO DO PROJETO” .....	85
<b>5.4.1</b>	<b>Tarefa “Realizar Reunião de Status do NPI” .....</b>	<b>85</b>
<b>5.4.2</b>	<b>Tarefa “Realizar Reunião Diária” .....</b>	<b>86</b>



5.5	ATIVIDADE “GERENCIAMENTO DE CONFIGURAÇÃO” .....	87
<b>5.5.1</b>	<b>Tarefa “Criar Plano de Gerenciamento de Configuração” .....</b>	<b>88</b>
<b>5.5.2</b>	<b>Tarefa “Estabelecer um Sistema de Gestão de Configuração” .....</b>	<b>89</b>
5.6	ATIVIDADE “ENCERRAR PROJETO” .....	90
<b>5.6.1</b>	<b>Tarefa “Elaborar Relatório Final” .....</b>	<b>90</b>
<b>5.6.2</b>	<b>Tarefa “Realizar Reunião de Encerramento do Projeto” .....</b>	<b>91</b>
5.7	CONCLUSÃO DA SEÇÃO .....	92
6	IMPLANTAÇÃO DO PROCESSO NO NPI.....	93
6.1	CHECKLIST DE ADERÊNCIA AO PROCESSO PROPOSTO.....	93
6.2	ANÁLISE DAS ENTREVISTAS.....	100
6.3	DIFICULDADES .....	102
6.4	LIÇÕES APRENDIDAS.....	102
6.5	OPORTUNIDADES DE MELHORIA .....	103
6.6	CONCLUSÃO DA SEÇÃO .....	104
7	CONCLUSÕES E TRABALHOS FUTUROS .....	105
7.1	CONSIDERAÇÕES FINAIS .....	105
7.2	TRABALHOS FUTUROS .....	106
	REFERÊNCIAS .....	109
	APÊNDICES .....	113
	APÊNDICE A - CHECKLIST DE ADERÊNCIA DO PROCESSO DO NPI.....	115
	APÊNDICE B - CASO DE USO CRUD GENÉRICO.....	123
	APÊNDICE C - CASO DE USO RELATÓRIO GENÉRICO.....	131
	ANEXOS .....	139
	ANEXO A - DOCUMENTO DE VISÃO.....	141
	ANEXO B - DOCUMENTO DE ESPECIFICAÇÃO DOS REQUISITOS.....	151



## 1 INTRODUÇÃO

A qualidade de um software depende da qualidade do processo e do produto, porém para se obter qualidade em um produto é imprescindível ter qualidade no processo que rege a sua construção. As empresas que constroem software estão em busca de se adequar a um mercado cada vez mais exigente com a qualidade dos produtos.

Com o objetivo de produzir software mais rápido e de maior qualidade as organizações estão em busca de definir e melhorar seus processos. “Um processo de software pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software” (FALBO; BARCELLOS, 2011, p. 5).

A definição de um processo visa à melhoria da qualidade dos produtos desenvolvidos por uma organização (FALBO; BARCELLOS, 2011, p. 6). A padronização é uma das maneiras de alcançar este objetivo e esta define como as atividades devem ser realizadas. Além disto, existem diversos outros benefícios em se utilizar processo, como:

- utilizar melhores práticas na produção do software, que podem ser guiados por modelos, guias, ferramentas, entre outros;
- desenvolver software de maior qualidade e mais rápido;
- produzir conhecimento de como realizar as atividades, quais artefatos gerar;
- motivar os envolvidos a seguir o processo proposto, diminuindo assim os erros que poderiam ocorrer e até o retrabalho;
- promover maior satisfação do cliente;
- possibilitar certificação através de organizações certificadoras que os produtos desenvolvidos seguem um padrão de qualidade, definido por estas;
- a obtenção da certificação tende a deixar a organização mais competitiva no mercado.

Para definir e melhorar seus processos, as organizações procuram seguir modelos, guias e metodologias para aperfeiçoar a construção e o desenvolvimento de software. Dentre os mais conhecidos, pode-se citar o Modelo Integrado de Maturidade e de Capacidade (*Capability Maturity Model Integration – CMMI*) (SEI, 2010), o modelo de Melhoria de Processo de Software Brasileiro (MPS.BR) (SOFTEX, 2011), o guia do Conhecimento em Gerenciamento de Projetos (*Project Management Body of Knowledge – PMBoK*) (PMI, 2008), a metodologia tradicional Processo Unificado da Rational (*Rational Unified Process –*

RUP) (WTHREEX, 2002) e a metodologia ágil Scrum (SCHWABER; SUTHERLAND, 2011).

Para mostrar os resultados obtidos das organizações que implementaram o modelo MPS.BR, um estudo foi realizado por Kalinowski *et al.* (2010), nos anos de 2008 e 2009. O estudo mostrou que as empresas obtiveram maior satisfação dos seus clientes, maior produtividade, aumento dos projetos, capacidade de desenvolver projetos maiores, comparadas com organizações que estão iniciando a implementação do MPS.BR. “Em 2008, 94,4% das organizações relataram estar totalmente (70,2%) ou parcialmente satisfeitas (24,2%) com o modelo. Em 2009, grande maioria das organizações (98,5%) relatou estar totalmente (71,1%) ou parcialmente satisfeita (27,4%)” (KALINOWSKI *et al.*, 2010, p. 11).

Diante desses resultados e de muitos outros, algumas organizações no ambiente acadêmico estão começando a enxergar os benefícios que um processo pode trazer. Pois, mesmo em projetos que sejam desenvolvidos dentro de universidades, é importante que exista uma formalização das atividades que são desenvolvidas e sejam utilizadas melhores práticas de processos, de forma que se obtenham os benefícios já constatados pelas empresas.

Algumas universidades já estão utilizando processos de software em seus projetos, como o Centro de Recursos Computacionais (CERCOMP) (MENDES *et al.*, 2010) e o Laboratório de Aplicações Ubíquas e Pervasivas (LUPA) (MENDES; ALMEIDA; JUNIOR, 2011), ambos da Universidade Federal de Goiás (UFG) e outras já foram avaliadas oficialmente no MPS.BR, como o Centro de Computação da Aeronáutica de São José dos Campos (CCA SJ) (SCHEID *et al.*, 2007) e o Laboratório de Engenharia de Software (LENS) da Coordenação de Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro (COPPE/UFRJ) (SANTOS *et al.*, 2009).

Porém, apesar de existirem instituições ligadas a universidades que implantaram processo e outras que foram avaliadas por modelos de melhoria de processos, não localizamos na literatura um trabalho aplicado em um Núcleo de Práticas voltado para o desenvolvimento de software por alunos de cursos de graduação em tecnologia da informação.

Neste contexto, este trabalho propôs a definição e implantação de um processo para o Núcleo de Práticas de Informática da Universidade Federal do Ceará – Campus Quixadá. O Núcleo de Práticas de Informática (NPI) foi criado no início da instalação da Universidade Federal do Ceará (UFC) no município de Quixadá, em 2008, com o objetivo de suprir as necessidades de sistemas para uso interno do campus. Porém, com sua evolução percebeu-se outras utilizações para ele, como o provimento de estágio para estudantes de graduação dos cursos ministrados na UFC – Campus Quixadá.

Como principal motivação do incentivo ao crescimento do NPI, pode-se destacar a baixa absorção dos alunos de graduação no mercado de desenvolvimento de software da cidade de Quixadá, devido a existirem poucas empresas instaladas voltadas para a área de tecnologia da informação, e o crescimento da demanda de software por parte dos parceiros do NPI. Além disso, o Núcleo de Práticas de Informática promove a prática na profissão e formação de mão de obra qualificada para o mercado de trabalho que está se instalando em Quixadá.

A definição e implantação de um processo de desenvolvimento de software no NPI tem como propósito, estabelecer um processo comum, padronizar os modelos de documentos e incorporar melhores práticas de modelos de qualidade de software.

## **1.1 Objetivos**

Este trabalho teve como principal objetivo definir e implantar um processo de desenvolvimento de software para o Núcleo de Práticas de Informática da Universidade Federal do Ceará do campus Quixadá, adequada a boas práticas de modelos de melhoria de processo, guias e metodologias ágeis.

Os objetivos específicos desse trabalho foram:

- capturar e analisar o desenvolvimento das atividades no Núcleo de Práticas de Informática;
- selecionar melhores práticas de modelos de processos de software tradicionais e ágeis como CMMI, MPS.BR, RUP, PMBoK, Scrum e SCORE. Realizar pesquisa de trabalhos de definição e implantação de processos em ambiente empresarial e acadêmico;
- definir um processo baseado nas boas práticas dos modelos e metodologias estudados, e na análise das atividades desenvolvidas no NPI, focando nas disciplinas de Gerência de Projetos, Requisitos e Gerência de Configuração;
- implantar o processo em projetos pilotos do NPI;
- coletar resultados da implantação do processo.

## **1.2 Estrutura do trabalho**

Este trabalho está estruturado em seis seções, além desta introdução.

A Seção 2, Fundamentação Teórica, apresenta os conceitos relevantes para o entendimento deste trabalho e contém a definição de processo, os modelos de processos de software utilizados neste trabalho, bem como processos de software definidos e implantados no ambiente empresarial e acadêmico.

A Seção 3, Procedimentos Metodológicos, apresenta a metodologia utilizada para a execução do trabalho.

A Seção 4, Definição do Processo do NPI, apresenta as fases relacionadas à construção e implantação do processo.

A Seção 5, Processo do NPI, apresenta detalhadamente todas as atividades e tarefas do processo proposto.

A Seção 6, Implantação do Processo no NPI, apresenta a implantação do processo e os seus resultados.

Por fim, a Seção 7, Conclusões e Trabalhos Futuros, apresenta as considerações finais acerca do trabalho realizado e propostas de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

A Seção Fundamentação Teórica aborda todos os conceitos relevantes para o entendimento do trabalho que foi desenvolvido. Esta Seção é composta de quatro subseções. A subseção 2.1, Processo de Desenvolvimento de Software, apresenta algumas definições de processo de software e seus benefícios. A subseção 2.2, Modelos de Melhoria de Processos de Software, apresenta todos os modelos de melhoria de processo de software utilizados neste trabalho. A subseção 2.3, Modelo de Gerenciamento de Projetos, apresenta o guia de gestão de projetos utilizado neste trabalho. A subseção 2.4, Metodologias de Desenvolvimento de Software, apresenta as metodologias tradicional e ágil utilizadas neste trabalho. A subseção 2.5, Definição e Implantação de Processos de Software, apresenta alguns trabalhos que definiram e implantaram processo de software no ambiente empresarial e acadêmico. A subseção 2.6, Conclusões, apresenta as considerações finais desta Seção.

### 2.1 Processo de Desenvolvimento de Software

“A Engenharia de Software evoluiu significativamente nas últimas décadas procurando estabelecer técnicas, critérios, métodos e ferramentas para apoiar a produção de software.” (BARBOSA *et al.*, 2000, p. 2). Apesar de todo esse apoio, ainda podem ocorrer erros nas atividades do processo de desenvolvimento de software. Com o intuito de garantir a qualidade em software, a Engenharia de Software possui uma subárea chamada de Qualidade de Software. Esta subárea trata da qualidade dos processos e dos produtos de software.

A qualidade do processo visa acompanhar todo o processo de desenvolvimento e inspecioná-lo com o intuito de diminuir o máximo de não conformidades que possam existir nele e a qualidade do produto visa acompanhar a construção do produto e retirar o máximo de defeitos possível para que o software tenha menos erros e conseqüentemente mais qualidade. Essas duas subáreas da Qualidade de Software se complementam, porém a qualidade do produto depende diretamente da qualidade do processo. Por isso existe uma preocupação das organizações em definir e implantar processos de software. Nesse contexto, buscando melhorar a qualidade dos produtos desenvolvidos pelo Núcleo de Práticas de Informática, este trabalho tem como objetivo definir e implantar um processo de software para esta organização.

### 2.1.1 Definição de Processo de Software

Existem diversas definições de processo de software e algumas delas são expressas a seguir. Processo é:

- “um conjunto de atividades inter-relacionadas ou interativas, que transforma insumos (entradas) em produtos (saídas)” (ABNT, 2001 *apud* SOFTEX, 2011, p. 10);
- “o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software” (FALBO; BARCELLOS, 2011, p. 5);
- “um conjunto de atividades relacionadas que levam à produção de um produto de software” (SOMMERVILLE, 2011, p. 18);
- “um conjunto de passos parcialmente ordenados com a intenção de atingir uma meta” (WTHREEX, 2002).

Em síntese, um processo de software é um conjunto de atividades, métodos, práticas que guiam as pessoas na produção de software, levando-as a produzir um produto como saída.

Segundo Falbo e Barcellos (2011, p. 5), um processo para ser eficaz deve relacionar a correlação entre as atividades, artefatos produzidos no desenvolvimento, as ferramentas utilizadas, bem como a motivação das pessoas envolvidas e suas habilidades.

O processo impacta fortemente na qualidade do produto final, pois o produto é construído baseado nele. Um processo de qualidade é muito importante para a qualidade do produto final, pois se o processo foi definido baseado em boas práticas de modelos de melhoria de processo de software e ele é seguido, o produto tende a ter uma qualidade muito maior do que se ele não fosse baseado em um processo.

O processo visa padronizar as atividades relativas a todas as fases da construção do software. Cada atividade desenvolvida possui diversos itens a serem cumpridos, como pré-atividades e pós-atividades, que são atividades que precedem e sucedem uma determinada atividade, as tarefas da atividade a serem desenvolvidas, seus artefatos, recursos necessários para a sua realização, bem como procedimentos que devem ser utilizados, que podem ser métodos, técnicas e modelos de documentos (FALBO; BARCELLOS, 2011, p. 5). Além de padronizar as atividades, o processo de software possui diversos outros benefícios para a organização que o utiliza, como:

- produz conhecimento sobre o desenvolvimento de software da organização;

- documenta as atividades que envolvem a produção do software, facilitando assim o repasse das informações referentes à execução das atividades aos novos integrantes do projeto;
- diminui o retrabalho, pois no processo estará muito bem detalhado a forma de execução das atividades, as ferramentas que devem ser utilizadas e os participantes envolvidos;
- possibilita estimativas mais precisas de tempo, custo e escopo;
- aumenta a qualidade do produto, pois se o processo está bem definido e é seguido, conseqüentemente influenciará que o produto gerado terá maior qualidade;
- aumenta a produtividade;
- reduz o tempo para atender o mercado, ou seja, mais rapidez na entrega do software;
- aumenta a competitividade no mercado.

Para se obter esses e outros benefícios é preciso definir um processo baseando-se em modelos de melhoria de processo de software conhecidos e utilizados pelo mercado. Neste trabalho foram selecionados dois modelos de melhoria de processos, o Modelo Integrado de Maturidade e de Capacidade (CMMI) e o modelo de Melhoria de Processo de Software Brasileiro (MPS.BR), além desses modelos foram selecionados, um guia do Conhecimento em Gerenciamento de Projetos (PMBok) e duas metodologias, a metodologia tradicional *Rational Unified Process* (RUP) e metodologia ágil Scrum. Estes são descritos na Seção abaixo.

## **2.2 Modelos de Melhoria de Processos de Software**

A definição do processo de software deve estar de acordo com as práticas utilizadas no processo de desenvolvimento da organização e pode se utilizar de modelos de melhoria de processo de software. Estes modelos descrevem um caminho para a melhoria de processos de software e sugerem boas práticas para serem utilizadas no processo. Dentre os modelos mais conhecidos temos o Modelo Integrado de Maturidade e de Capacidade (CMMI) e o modelo de Melhoria de Processo de Software Brasileiro (MPS.BR). Além destes modelos, em diversos trabalhos (CATUNDA *et al.*, 2011; SILVA *et al.*, 2011; CINTRA, 2006; SALGADO *et al.*, 2010) utilizam-se desses modelos juntamente com metodologias tradicionais e ágeis e guia em gerenciamento de projetos em conjunto, com o intuito de

enriquecer ainda mais o processo de software que está sendo contruído. Com isso, esse trabalho optou por utilizar práticas dos modelos de melhoria de processo mais conhecidos no mercado brasileiro que são o CMMI e o MPS.BR, o guia em gerenciamento de projetos PMBoK, as metodologias tradicional RUP e ágil Scrum, e o SCORE, adaptação do Scrum para grupos de pesquisa. Eles foram escolhidos por já terem sido utilizados em diversos trabalhos (CATUNDA *et al.*, 2011; CINTRA, 2006; SALGADO *et al.*, 2010) e alguns serem realizados por instituições implementadoras do modelo MPS.BR (SANTOS *et al.*, 2009; SCHEID *et al.*, 2007), além de terem tido ótimos resultados.

Os modelos citados acima e utilizados neste trabalho são descritos a seguir.

### **2.2.1 Modelo Integrado de Maturidade e de Capacidade (CMMI)**

O Instituto de Engenharia de Software (*Software Engineering Institute – SEI*) criou diversos modelos para melhoria de processo, dentre eles o SW-CMM e o CMMI.

O Modelo de Maturidade e de Capacidade para Software (*Capability Maturity Model for Software – SW-CMM*) foi criado no final da década de 1980, especificamente para software, patrocinado pelo Departamento de Defesa dos EUA. Ele é um modelo de capacitação de processo que foi criado para a avaliação da capacidade dos fornecedores de software do seu patrocinador. Seu objetivo principal é “que as organizações conheçam e melhorem seus processos de desenvolvimento de software com a implementação de práticas definidas.” (KOSCIANSKI; SOARES, 2007, p. 95).

A partir do sucesso do SW-CMM foram criados novos modelos semelhantes a este para diversas áreas, como gestão de recursos humanos (P-CMM), de aquisição de software (SA-CMM) e de engenharia de sistemas (SE-CMM). Porém, esses modelos apresentavam diversas estruturas, formatos e termos diferentes, o que causava confusão quando se precisava utilizar mais de um deles. “Com a finalidade de integrar os diversos modelos criados e como uma evolução do Modelo de Maturidade e de Capacidade (*Capability Maturity Model – CMM*)”, e também com o objetivo de diminuir a redundância e eliminar a inconsistência que surgia ao se utilizar diversos modelos independentes, foi criado o Modelo Integrado de Maturidade e de Capacidade (*Capability Maturity Model Integration – CMMI*) (KOSCIANSKI; SOARES, 2007, p. 102).

O CMMI tem como objetivo “servir de guia para a melhoria de processos na organização e também da habilidade dos profissionais em gerenciar o desenvolvimento, aquisição e manutenção de produtos e serviços.” (KOSCIANSKI; SOARES, 2007, p. 102).

Segundo Barbosa, Furtado e Gomes (2006, p. 2 *apud* GABRIEL, 2009, p. 34), o CMMI é “um modelo de melhoria de processos que serve como opção para as organizações, quando elas desejam tornar mais eficiente a gerência de seus projetos ao considerar, entre outros aspectos, o custo, o tempo e os recursos utilizados”.

O CMMI é dividido em quatro corpos de conhecimento ou disciplinas, que são Engenharia de Sistemas, Engenharia de Software, Desenvolvimento Integrado do Produto e do Processo e Fontes de Aquisição e possui dois tipos de representação, a contínua e por estágios.

A representação por estágios permite as organizações uma melhoria gradual dos processos através de áreas de processo e utiliza níveis de maturidade para caracterizar o estado geral dos processos da organização (SEI, 2010, p. 22, tradução nossa). Os níveis de maturidade agrupam áreas de processo que são necessárias para se atingir uma determinada maturidade.

Essa representação é composta por cinco níveis de maturidade que são, nível 1 (Inicial), 2 (Gerenciado), 3 (Definido), 4 (Gerenciado quantitativamente) e 5 (Otimizado). Os níveis sugerem uma ordem para a melhoria dos processos. Em cada um destes níveis, existem áreas de processo predefinidas que precisam ser atingidas para se obter um determinado nível de maturidade. Para se atingir um nível, todas as áreas de processo deste nível e dos anteriores a este devem ser satisfeitas. O Quadro 1 contém os níveis de maturidade e as respectivas áreas de processos que devem ser atingidos em cada nível.

A representação contínua “permite que as organizações melhorem um conjunto de processos relacionados, de forma incremental abordando conjuntos sucessivos de áreas de processo.” (SEI, 2010, p. 21, tradução nossa). Nesse tipo de representação a organização pode selecionar as áreas de processo as quais ela desejar ter um nível de capacidade. Cada área de processo é avaliada individualmente e atinge um nível de capacidade individual, ou seja, diversas áreas de processo podem ter níveis de capacidade diferentes.

Essa representação estabelece quatro níveis de capacitação que são, nível 0 (Incompleto), 1 (Realizado), 2 (Gerenciado) e 3 (Definido). “Nesta representação, as áreas de processo são agrupadas por categorias afins.” (KOSCIANSKI; SOARES, 2007, p. 109). As áreas de processo do CMMI para software na representação contínua são agrupadas em quatro categorias: Gerência de processos, Gerência de projeto, Engenharia e Suporte.

Quadro 1 – Áreas de processo – CMMI por estágios

Níveis de Maturidade	Áreas de Processo
Nível de Maturidade 2	Garantia da Qualidade de Processo e Produto – PPQA
	Gerência de Configuração – CM
	Gestão de Contrato com Fornecedores – SAM
	Gestão de Requisitos – REQM
	Medição e Análise – MA
	Monitoramento e Controle do Projeto – PMC
	Planejamento de Projeto – PP
Nível de Maturidade 3	Análise e Tomada de Decisões – DAR
	Definição dos Processos da Organização – OPD
	Desenvolvimento de Requisitos – RD
	Foco nos Processos da Organização – OPF
	Gestão de Riscos – RSKM
	Gestão Integrada de Projeto – IPM
	Integração de Produto – PI
	Solução Técnica – TS
	Treinamento da Organização – OT
	Validação – VAL
	Verificação – VER
Nível de Maturidade 4	Desempenho dos Processos da Organização – OPP
	Gestão Quantitativa de Projeto – QPM
Nível de Maturidade 5	Análise e Resolução de Causas – CAR
	Gestão do Desempenho da Organização – OPM

Fonte: adaptado de SEI (2010, p. 33-34).

### 2.2.2 Modelo de Melhoria de Processo de Software Brasileiro (MPS.BR)

O modelo de Melhoria de Processo de Software Brasileiro (MPS.BR) visa a definição, aprimoramento e avaliação do processo de software das micro, pequenas e médias empresas, mas ele também pode ser utilizado em grandes empresas, apesar de seu foco não ser nesta última (SOFTEX, 2011, p. 12).

A sua construção foi baseada nas normas ISO/IEC 12207:2008 e ISO/IEC 15504-2 e no Modelo Integrado de Maturidade e de Capacidade para Desenvolvimento (*Capability Maturity Model Integration for Development – CMMI-DEV*) (SOFTEX, 2011, p. 13). “A norma ISO/IEC 15504 apresenta uma estrutura para a realização de avaliações de processo em organizações.” (KOSCIANSKI; SOARES, 2007, p. 156). Já a norma ISO/IEC 12207 “provê uma estrutura para que uma organização defina os seus processos.” (KOSCIANSKI; SOARES, 2007, p. 164). O modelo CMMI tem como objetivo servir de guia para melhoria de processos nas organizações e também auxiliar os profissionais na aquisição e manutenção de produtos ou serviços (KOSCIANSKI; SOARES, 2007, p. 102).

O modelo MPS.BR está dividido em três componentes, que são Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS). O Modelo de Referência MR-MPS contém os requisitos que devem ser atendidos pelas

organizações para estar em conformidade com o MR-MPS. Nele estão descritos as definições dos níveis de maturidade, processos e atributos do processo (SOFTEX, 2011, p. 13). “O Modelo de Avaliação contém o processo de avaliação, os requisitos para os avaliadores e os requisitos para averiguação da conformidade ao modelo MR-MPS.” (KOSCIANSKI; SOARES, 2007, p. 144). “O Modelo de Negócio contém uma descrição das regras para a implementação do MR-MPS pelas empresas de consultoria, de software e de avaliação.” (KOSCIANSKI; SOARES, 2007, p. 144).

Dentre estes modelos o que contém a descrição dos requisitos que precisam ser atendidos é o Modelo de Referência MR-MPS. Ele define níveis de maturidade. “O nível de maturidade em que se encontra uma organização permite prever seu desempenho futuro.” (KOSCIANSKI; SOARES, 2007, p. 144).

Os níveis de maturidade são uma combinação entre processos e sua capacidade. A definição do processo declara o seu propósito e os resultados esperados na sua execução. Já a capacidade de processo é “uma caracterização da habilidade do processo atingir aos objetivos de negócio atuais ou futuros.” (SOFTEX, 2011, p. 8-16).

O MR-MPS possui sete níveis de maturidade, os quais devem permitir uma implementação e avaliação em prazos mais curtos e mais adequada as micro, pequenas e médias empresas. Os sete níveis de maturidade são A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). Os níveis de maturidade se iniciam no nível G e progridem até o nível A. “Para cada um destes sete níveis de maturidade é atribuído um perfil de processos que indicam onde a organização deve colocar o esforço de melhoria.” (SOFTEX, 2011, p. 16). O progresso e o alcance de um determinado nível de maturidade são obtidos quando os processos e capacitação dos processos são alcançados naquele determinado nível e nos níveis anteriores a este. O Quadro 2 contém os níveis e os respectivos processos que devem ser atingidos em cada nível.

### **2.3 Modelo de Gerenciamento de Projetos**

Os projetos desenvolvidos por uma organização precisam ser gerenciados para serem executados da melhor forma possível. O gerenciamento de um projeto tem como objetivo controlá-lo fazendo com que ele consiga ser finalizado dentro do tempo, escopo e custo estimados. Mas, para isso, os integrantes dos projetos precisam ter conhecimentos de

como gerenciar projetos. Esses conhecimentos podem ser encontrados no guia de Conhecimento em Gerenciamento de Projetos (PMBok), o qual é apresentado a seguir.

Quadro 2 – Níveis de maturidade do MR-MPS

Nível	Processos
A	(sem processos adicionais)
B	Gerência de Projetos – GPR (evolução)
C	Gerência de Riscos – GRI Desenvolvimento para Reutilização – DRU Gerência de Decisões – GDE
D	Verificação – VER Validação – VAL Projeto e Construção do Produto – PCP Integração do Produto – ITP Desenvolvimento de Requisitos – DRE
E	Gerência de Projetos – GPR (evolução) Gerência de Reutilização – GRU Gerência de Recursos Humanos – GRH Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional – AMP
F	Medição – MED Garantia da Qualidade – GQA Gerência de Portfólio de Projetos – GPP Gerência de Configuração – GCO Aquisição – AQU
G	Gerência de Requisitos – GRE Gerência de Projetos – GPR

Fonte: adaptado de SOFTEX (2011, p. 22).

### 2.3.1 Guia do Conhecimento em Gerenciamento de Projetos (PMBok)

O Guia do Conhecimento em Gerenciamento de Projetos (*Project Management Body of Knowledge* – PMBoK) é desenvolvido pelo *Project Management Institute* (PMI). Ele é um guia de melhores práticas e conhecimento em gerenciamento de projetos que fornece diretrizes para o gerenciamento de projetos individuais, define o gerenciamento e os conceitos relacionados, descreve o ciclo de vida do gerenciamento de projetos e os processos relacionados (PMI, 2008, p. 10).

O guia PMBoK possui 42 processos, divididos em 9 áreas de conhecimento e 5 grupos de processos.

As áreas de conhecimento do PMBoK caracterizam os principais aspectos envolvidos em um projeto e no seu gerenciamento. As nove áreas são Integração, Escopo, Tempo, Custos, Qualidade, Recursos Humanos, Comunicação, Riscos e Aquisições.

Segundo Márcio D'ávila (2006), as áreas de conhecimento Escopo, Tempo, Custos e Qualidade são os principais determinantes para o objetivo de um projeto. Essas áreas concentram as preocupações centrais dos projetos que são entregar um resultado de acordo

com o escopo, no prazo e no custo definidos, com a qualidade adequada. As áreas de conhecimento Recursos Humanos e Aquisições são as entradas para se produzir o trabalho do projeto. As áreas de conhecimento Comunicações e Riscos tratam de manter as expectativas dos *stakeholders* e as incertezas do projeto sob controle. E a área de conhecimento “Integração abrange a orquestração de todos estes aspectos” (D’ÁVILA, 2006).

A área de conhecimento Integração é composta pelos “processos e as atividades necessárias para identificar, definir, combinar, unificar e coordenar os vários processos e atividades dos grupos de processos de gerenciamento.” (PMI, 2008, p. 67).

A área de conhecimento Escopo abrange processos de gerenciamento e tem como objetivo garantir que o projeto realizará todo e somente o trabalho necessário para que o projeto seja bem-sucedido (DINSMORE, CAVALIERI, 2011, p. 49).

A área de conhecimento Tempo é composta por processos necessários para que o projeto chegue ao término no prazo correto.

A área de conhecimento Custos é composta por processos responsáveis por realizar estimativas, orçamentos e controle dos custos, visando assegurar que o projeto possa ser terminado dentro do orçamento aprovado.

A área de conhecimento Qualidade é composta pelos processos responsáveis por assegurar que o projeto seja concluído dentro da qualidade desejada, e que dessa forma, garanta a satisfação dos clientes (DINSMORE, CAVALIERI, 2011, p. 135).

A área de conhecimento Recursos Humanos é composta pelos processos relacionados à organização e gerenciamento da equipe de projeto (PMI, 2008, p. 181).

A área de conhecimento Comunicação é composta pelos processos que asseguram que as informações dos projetos sejam geradas, coletadas, distribuídas, armazenadas, recuperadas e organizadas de maneira apropriada (PMI, 2008, p. 204).

A área de conhecimento Riscos é composta pelos processos relacionados ao planejamento, identificação, análise, planejamento de respostas, monitoramento e controle de riscos de um projeto (PMI, 2008, p. 226).

A área de conhecimento Aquisições é composta pelos processos relacionados a compras ou aquisição de produtos, serviços ou resultados externos à equipe do projeto (PMI, 2008, p. 259).

Cada área de conhecimento abrange diversos processos. Cada processo é detalhado e contém a sua descrição, as entradas, saídas, ferramentas e técnicas a serem utilizadas, e cada processo está relacionado a um grupo de processos.

Os grupos de processos do guia PMBoK são Iniciação, Planejamento, Execução, Monitoramento e Controle, e Encerramento (PMI, 2008, p. 39).

O grupo de processos de Iniciação é composto pelos “processos realizados para definir um novo projeto ou uma nova fase de um projeto existente através da obtenção de autorização para iniciar o projeto ou a fase.” (PMI, 2008, p. 39).

O grupo de processos de planejamento é composto pelos “processos realizados para definir o escopo do projeto, refinar os objetivos e desenvolver o curso de ação necessário para alcançar os objetivos para os quais o projeto foi criado.” (PMI, 2008, p. 39).

O grupo de processos de execução tem como finalidade coordenar as pessoas e os recursos para que o projeto seja executado de acordo com o plano de gerenciamento do projeto.

O grupo de processos de monitoramento e controle é composto pelos processos necessários para acompanhar o andamento do projeto, por verificar se a execução do projeto está dentro do prazo, custo e escopo especificados, se o produto está sendo produzido com a qualidade esperada. Os processos desse grupo concentram-se no monitoramento e na mensuração do desempenho do projeto para identificar variações no seu plano e realinhamento com este.

O grupo de processos de encerramento é composto pelos “processos executados para finalizar todas as atividades de todos os grupos de processos, visando encerrar formalmente o projeto ou a fase.” (PMI, 2008, p. 39).

## **2.4 Metodologias de Desenvolvimento de Software**

As metodologias de desenvolvimento de software descrevem algumas práticas que devem ser seguidas para se ter um bom desenvolvimento de software. A escolha da metodologia depende de diversos fatores, como o tipo do projeto, o seu tamanho e o seu escopo. As metodologias podem ser tradicional ou ágil. Neste trabalho, optou-se por combinar as duas metodologias com o intuito de extrair o melhor delas. As metodologias utilizadas neste trabalho são explanadas a seguir.

### 2.4.1 *Processo Unificado da Rational (RUP)*

O Processo Unificado da Rational (*Rational Unified Process* – RUP), também chamado de processo RUP, é um processo de engenharia de software criado pela *Rational Software Corporation* e adquirido posteriormente pela IBM em 2003.

O RUP é um processo de engenharia de software que tem como objetivo a produção de software com a mais alta qualidade atendendo as necessidades dos usuários, dentro de um prazo e custos previsíveis (KRUCHTEN, 2003, p. 15). O RUP por ser flexível e configurável, pode ser utilizado em projetos de diversos tamanhos, desde os de grande porte até os pequenos.

O RUP é constituído de melhores práticas, fases do ciclo de vida e cinco elementos principais.

As melhores práticas do RUP têm como intuito tentar diminuir os riscos do desenvolvimento de software. O RUP tem como melhores práticas, desenvolver o software iterativamente, gerenciar os requisitos, utilizar arquiteturas baseadas em componentes, modelar visualmente nos permitindo ter uma visão simplificada do sistema, verificar continuamente a qualidade objetivando encontrar e corrigir qualquer inconformidade que possa surgir e gerenciar as mudanças que possam ocorrer durante todo o ciclo de vida do projeto.

O RUP possui uma arquitetura geral, a qual é composta de duas dimensões: o eixo horizontal e o eixo vertical. O eixo horizontal representa o tempo e mostra os aspectos do ciclo de vida do processo à medida que se desenvolve e o eixo vertical representa as disciplinas, que agrupam as atividades de maneira lógica, por natureza (KRUCHTEN, 2003, p. 19). A Figura 1 mostra a arquitetura geral do RUP, com suas fases e disciplinas.

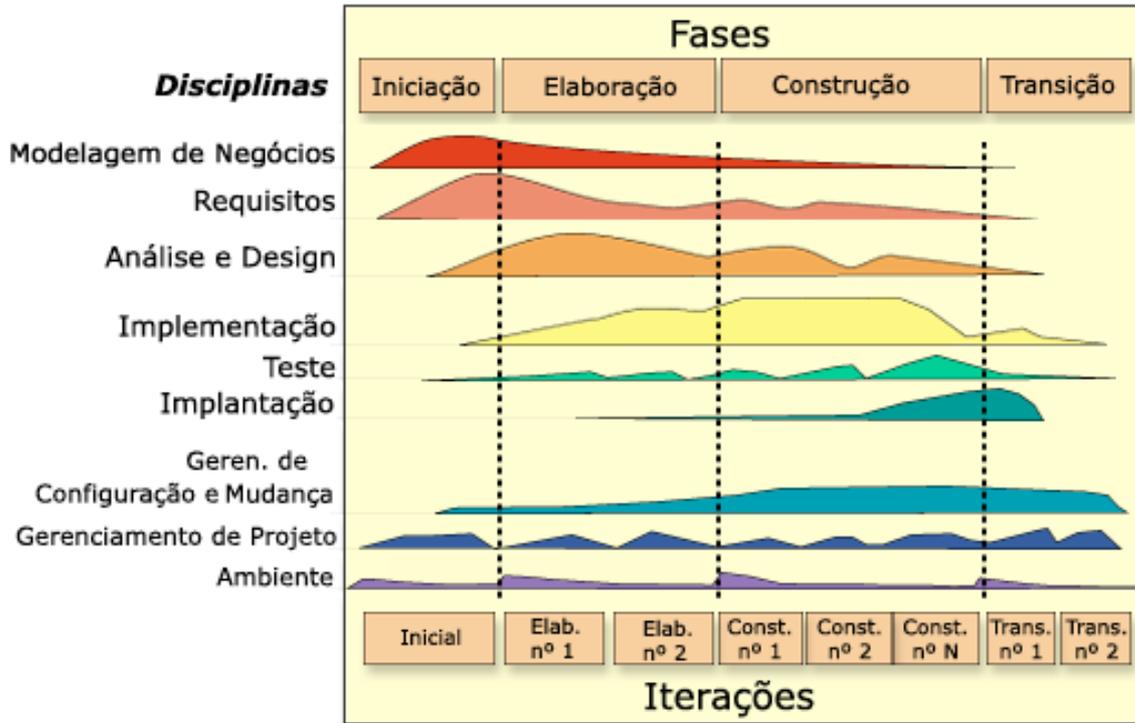
O ciclo de vida do RUP é dividido em quatro fases. Cada passagem pelas quatro fases consiste de uma iteração. As fases do RUP são Iniciação, Elaboração, Construção e Transição.

A fase de Iniciação tem como objetivo determinar o escopo do projeto, estimar os custos, o cronograma e os riscos.

A fase de Elaboração analisa o domínio do problema, desenvolve o plano de projeto, estabelece a base arquitetural e elimina os elementos de alto risco (PISKE, 2003). Nessa fase, o escopo deve ser revisado e os requisitos devem estar mais compreendidos. Esta é considerada a fase mais crítica, pois ao final desta fase a engenharia é considerada completa e esta é a hora de decidir se o projeto vai ou não para as fases de construção e transição

(BORK, 2003). Além disso, quanto mais avançado o projeto estiver maiores serão os custos com modificações.

Figura 1 - Arquitetura Geral do RUP



Fonte: WTHREEX (2002).

A fase de Construção tem como meta esclarecer os requisitos restantes e realizar o desenvolvimento do produto com base na arquitetura projetada na fase anterior. Nessa fase os recursos são gerenciados, o produto é desenvolvido, testado e deve ser integrado ao restante do sistema.

A fase de Transição tem como propósito realizar a transição da versão beta para uma versão que será utilizada por todos os usuários. Nessa fase o sistema é implantado e novas versões são lançadas com o intuito de corrigir problemas (BORK, 2003).

No RUP, além das fases e suas iterações, existem as disciplinas. Uma disciplina é uma coleção de atividades relacionadas que fazem parte de um contexto comum em um projeto. As disciplinas do RUP são Modelagem de Negócios, Requisitos, Análise e Design, Implementação, Teste, Implantação, Ambiente, Gerenciamento de Projeto e Gerenciamento de Configuração e Mudança. Essas disciplinas são partes clássicas e essenciais em qualquer projeto de desenvolvimento de software.

### 2.4.2 Scrum

“Scrum é um *framework* para desenvolver e manter produtos complexos.” Ele é fundamentado nas teorias empíricas de controle de processo, que “afirma que o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido”, e emprega a abordagem iterativa e incremental, com o intuito de aperfeiçoar a previsibilidade e controlar os riscos. Para implementar o controle de processo empírico, o Scrum é apoiado por três pilares: transparência, inspeção e adaptação (SCHWABER; SUTHERLAND, 2011, p. 3-4).

O primeiro pilar do Scrum é a Transparência, que diz que os aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados e que “requer aspectos definidos por um padrão comum para que os observadores compartilhem um mesmo entendimento do que está sendo visto” (SCHWABER; SUTHERLAND, 2011, p. 4).

O segundo pilar do Scrum é a Inspeção. Nele, os artefatos devem ser frequentemente inspecionados para que sejam detectadas possíveis variações em seus objetivos, porém a inspeção não deve ser tão frequente que chegue a atrapalhar a execução das tarefas. As inspeções são mais benéficas quando são executadas por pessoas especializadas no trabalho que será verificado.

O terceiro pilar do Scrum é a Adaptação, que diz que caso seja detectado que “um ou mais aspectos de um processo desviou-se para fora dos limites aceitáveis, e que o produto resultado será inaceitável, o processo ou o material sendo produzido deve ser ajustado” (SCHWABER; SUTHERLAND, 2011, p. 4).

O Scrum consiste em papéis, eventos, artefatos e regras. Cada componente serve a um propósito específico e é essencial para o uso e o sucesso do Scrum. “As suas regras integram os eventos, papéis e artefatos, administrando as relações e interações entre eles.” (SCHWABER; SUTHERLAND, 2011, p. 3). O Quadro 3 mostra um resumo das principais características da metodologia ágil Scrum.

No Scrum, toda a equipe que está envolvida com o projeto é denominada Time Scrum. Cada membro envolvido possui um papel. Os papéis do Scrum são *Product Owner*, a equipe de desenvolvimento e o *Scrum Master*.

O *Product Owner*, ou dono do produto, “é o responsável por maximizar o valor do produto e do trabalho da equipe de desenvolvimento.” (SCHWABER; SUTHERLAND, 2011, p. 5).

A equipe de desenvolvimento é o grupo de pessoas responsáveis por desenvolver as funcionalidades do produto. As equipes são auto-gerenciadas, auto-organizadas e multifuncionais. “O tamanho ideal da Equipe de Desenvolvimento é pequeno o suficiente para se manter ágil e grande o suficiente para completar uma parcela significativa do trabalho.” (SCHWABER; SUTHERLAND, 2011, p. 6).

O *Scrum Master* é o responsável por garantir que o Scrum seja entendido e aplicado.

Quadro 3 – Resumo das características da metodologia ágil Scrum

Resumo da metodologia ágil Scrum	
Papéis	<i>Product Owner</i>
	Equipe de desenvolvimento
	<i>Scrum Master</i>
Eventos	<i>Sprint</i>
	Reunião de Planejamento da <i>Sprint</i>
	Reunião Diária
	Revisão da <i>Sprint</i>
	Retrospectiva da <i>Sprint</i>
Artefatos	<i>Backlog</i> do Produto
	<i>Backlog</i> da <i>Sprint</i>
	Incremento

Fonte: a própria autora.

Além dos papéis, o Scrum prescreve eventos como forma de criar uma rotina de trabalho e minimizar a necessidade de reuniões não definidas previamente. Seus eventos têm um tempo de duração máxima, denominados *time-box*. Os eventos do Scrum são *Sprint*, Reunião de Planejamento da *Sprint*, Reunião Diária, Revisão da *Sprint* e Retrospectiva da *Sprint*.

O coração do Scrum é a *Sprint*, que é um *container* para todos os outros eventos. Ela dura em média um mês ou menos e é composta do desenvolvimento das funcionalidades selecionadas no *Backlog* do Produto, por uma reunião de planejamento da *Sprint*, reuniões diárias, uma revisão da *Sprint* e a retrospectiva da *Sprint*. (SCHWABER; SUTHERLAND, 2011, p. 8).

A reunião de planejamento da *Sprint* é a reunião onde o trabalho a ser realizado na *Sprint* é definido. Ela tem duração de oito horas para uma *Sprint* de um mês. “Para *Sprints* menores, este evento deve ser proporcionalmente menor.” (SCHWABER; SUTHERLAND, 2011, p. 9). A reunião é feita de maneira colaborativa por todo o time. Ela consiste de duas partes e cada uma delas dura metade do tempo da reunião, e respondem as seguintes questões (SCHWABER; SUTHERLAND, 2011, p. 9):

- O que será entregue como resultado do incremento da próxima *Sprint*?
- Como o trabalho necessário para entregar o incremento será realizado?

A reunião diária do Scrum é uma reunião com duração de 15 minutos, com o propósito de inspecionar o trabalho realizado desde a última reunião diária, prever o que será feito antes da próxima reunião e conhecer os obstáculos que estão impedindo das atividades serem realizadas. Ela deve ocorrer no mesmo local e horário todo dia para diminuir a complexidade.

A revisão da *Sprint* ocorre no final de cada *Sprint* e tem o propósito de inspecionar o trabalho realizado e mudar o *Backlog* do Produto, caso seja necessário. É uma reunião com 4 horas de duração para uma *Sprint* de um mês. Caso a *Sprint* seja menor, a duração da revisão da *Sprint* é proporcional ao tempo desta. O resultado desta reunião é um *Backlog* do Produto atualizado, pronto para o planejamento da *Sprint*.

A retrospectiva da *Sprint* ocorre depois da revisão da *Sprint* e antes da reunião de planejamento da próxima *Sprint*. Ela tem uma duração de 3 horas para uma *Sprint* de um mês. Ela é uma oportunidade que o time Scrum tem para inspecionar a si próprio e criar um plano de melhorias para a *Sprint* seguinte. Ao final da retrospectiva da *Sprint*, o time Scrum deve ter identificado melhorias que serão implementadas na próxima *Sprint*.

Dentro dos eventos do Scrum existem alguns artefatos que precisam ser desenvolvidos. “Os artefatos do Scrum representam o trabalho ou o valor das várias maneiras que são úteis no fornecimento de transparência e oportunidades para inspeção e adaptação.” (SCHWABER; SUTHERLAND, 2011, p. 12). Os artefatos do Scrum são *Backlog* do Produto, *Backlog* da *Sprint* e Incremento.

O *Backlog* do Produto (*Product Backlog*) é uma lista ordenada contendo os requisitos do software. Ele nunca está completo, sempre é atualizado ao decorrer do projeto. Ele lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas versões futuras. Nele os itens possuem atributos de descrição, ordem e estimativa. O *Backlog* do Produto acaba quando o projeto chega ao fim. O responsável por sua criação e atualização é o *Product Owner*.

O *Backlog* da *Sprint* é “um conjunto de itens do *Backlog* do Produto selecionados para a *Sprint*, juntamente com o plano de entrega do incremento do produto e atingir o objetivo da *Sprint*.” (SCHWABER; SUTHERLAND, 2011, p. 14).

O Incremento é a soma de todos os itens do *Backlog* do Produto completos na *Sprint* atual e nas anteriores.

### 2.4.3 *Scrum for Research (SCORE)*

O Scrum é uma metodologia ágil para projetos de desenvolvimento de produtos, porém existe um Scrum adaptado para grupos de pesquisa e se chama SCORE, que deriva das palavras “*SCrum fOr REsearch*”. As principais características do SCORE são (HICKS; FOSTER, 2010, p. 1-4):

- **reunião de *Status***: Baseia-se na ideia principal do Scrum, que são reuniões com *time-box* de 15 minutos, denominada de *status*. Ocorrem três vezes por semana, com todos os orientadores e seus alunos. O propósito da reunião de *status* é muito parecido com o do Scrum. Nela é relatado o progresso dos alunos em suas pesquisas, bem como os obstáculos encontrados e o planejamento dos próximos passos;
- **reunião técnicas sob demanda**: Diferentemente do Scrum possuem reuniões sobre demanda. Estas são agendadas de acordo com a necessidade individual de cada aluno. Ela tem como propósito discutir questões técnicas da pesquisa, desafios e resultados;
- para aumentar o espírito de grupo ainda mais, eles fazem um almoço semanal e possuem um grupo de leitura que se reúne um dia na semana, entre outras atividades.

## 2.5 Definição e Implantação de Processos de Software

A definição e implantação de um processo de software tem como objetivo estabelecer um processo padrão para uma organização padronizando os documentos, o modo de execução das atividades, atribuindo responsabilidades, entre outros. Porém, ela ainda é um desafio. Diversos fatores podem influenciar negativamente um programa de melhoria de processo de software. Por exemplo, os modelos de melhoria de processo existentes são voltados para organizações de médio e grande porte. Para serem utilizados em empresas brasileiras, eles tem de ser adaptados, além disso, existe ainda os altos custos da implantação destes modelos. Atualmente existe um modelo de melhoria de processo de software voltado para o mercado brasileiro, o MPS.BR, porém a sua implantação é algo que demanda tempo, dinheiro e recursos humanos. Além destes fatores, podemos citar o fator cultural e a resistência a mudanças. Apesar dos desafios, sabe-se que os benefícios alcançados com a

implantação podem ser bem maiores, principalmente quando ela é bem planejada e acompanhada adequadamente.

Para realizar a definição e implantação da melhor forma possível, alguns trabalhos foram estudados para agregar conhecimento da maneira de definir e implantar processo.

### **2.5.1 Processos de Software nas Empresas**

Várias empresas de desenvolvimento de software estão utilizando-se de processos com o intuito de melhorar a qualidade dos produtos desenvolvidos, padronizar as atividades e os documentos gerados. Para alcançar estas metas, as organizações estão procurando utilizar modelos de melhoria de processo com o objetivo de melhorar a qualidade do processo de desenvolvimento de seus produtos, bem como do produto final gerado e também para mostrar ao mercado consumidor que eles se preocupam com a qualidade. Com isso, diversos trabalhos de definição e implantação de processo vêm sendo desenvolvidos nas empresas. A seguir são apresentados alguns destes trabalhos.

#### **2.5.1.1 Serviço Federal de Processamento de Dados (SERPRO)**

No Serviço Federal de Processamento de Dados (SERPRO) foi definido e implantado um processo para a Gerência de Requisitos. A implantação do processo foi realizada nas regionais e em cada uma delas, foi escolhido um projeto para ser implantado. A equipe do projeto escolhido deveria disseminar conhecimento com as demais equipes do Polo (CARVALHO *et al.*, 2001).

Para a implantação do processo, nove fases deveriam ser seguidas até se conseguir o resultado esperado. Ao longo da implantação, os processos deveriam ser avaliados e ajustados de acordo com o *feedback* obtido. Os processos poderiam ser selecionados ou omitidos de acordo com as características e necessidades da organização. As nove fases da implantação do processo são explanadas no Quadro 4 (CARVALHO *et al.*, 2001).

#### **2.5.1.2 Rightway Consultoria & Sistemas**

A *Rightway Consultoria & Sistemas* é uma empresa com sede no Rio de Janeiro, especializada em desenvolvimento de soluções e projetos de Tecnologia da Informação (CATUNDA *et al.*, 2011).

Quadro 4 – Fases da implantação do processo do SERPRO

Fases	Descrição da fase
Conscientização	Nessa fase foi identificado a situação atual da organização, definido o status desejado e ações foram tomadas para diminuir a distância entre o status atual e o almejado. O status atual foi obtido por meio de observações, questionários e entrevistas com os desenvolvedores a respeito do conhecimento e da utilização dos Processos de Engenharia de Requisitos. As informações obtidas foram consolidadas e apresentadas aos gerentes e desenvolvedores.
Preparação	Nesta fase foram escolhidos grupos de trabalho, os quais estavam profundamente familiarizados com os processos de Engenharia de Requisitos. Um destes grupos, teve a tarefa de planejar e acompanhar todas as tarefas do projeto de melhoria.
Treinamento	Foi definido um Plano de Treinamento e priorizado a participação das pessoas que compõem os grupos de trabalho. O ideal é que o treinamento seja preparado e ministrado pelos integrantes do grupo que está planejando e acompanhando o projeto de melhoria. Durante o treinamento devem ser anotadas sugestões e considerações que possam surgir.
Levantamento de processos Engenharia de Requisitos existentes	Foram realizadas reuniões, entrevistas e coletas de documentos para se tomar conhecimento dos processos de Engenharia de Requisitos existentes na empresa. Através disso, o grupo identificou os pontos fortes e fracos.
Elaboração do Plano de Melhoria	Foi preparado o Plano de Melhoria contendo os processos e documentos existentes, onde há necessidade de melhoria e quais os benefícios que essas podem causar no processo.
Definição dos processos	O processo foi definido tendo como base as boas práticas já existentes na empresa, no Plano de Melhoria definido, nas restrições e na cultura da organização.
Definição de papéis e responsabilidades	Os papéis e responsabilidades das pessoas que executarão as atividades de desenvolvimento e gerenciamento de requisitos foram especificados e mapeados para as funções existentes na empresa.
Implantação	Foi realizado o planejamento da implantação e a execução do processo em um projeto piloto e posteriormente foi expandido para todos os projetos. Para iniciar a implantação foi definido um Plano para projeto piloto, onde foi especificado o projeto escolhido, as pessoas envolvidas, os treinamentos que serão necessários e os prazos esperados. A implantação do projeto piloto foi acompanhada e as dificuldades encontradas foram documentadas para posterior análise e avaliação dos processos definidos.
Acompanhamento e ajustes	O processo foi acompanhado, avaliado e ajustes foram realizados de acordo com a necessidade.

Fonte: CARVALHO *et al.* (2001).

Para definirem e implantarem processo de software eles utilizaram a seguinte estratégia. Primeiramente, ocorreu uma análise das práticas, processos e procedimentos da empresa. Esta análise, juntamente com as práticas ágeis já utilizadas na empresa, serviram como insumos para a definição da primeira versão do processo. Este foi instanciado em um projeto piloto.

A partir da execução do processo, a empresa pode identificar várias oportunidades de melhoria. Paralelo a execução do processo, ocorreu uma capacitação *Scrum Master*, onde a gerência da fábrica de software e todos os coordenadores de projetos fizeram o curso e se tornaram *Certified Scrum Masters*, aumentando assim o conhecimento da empresa sobre esta metodologia ágil. Com o maior conhecimento a respeito do Scrum e as oportunidades de melhoria identificadas na primeira versão processo, foram sugeridas alterações no processo.

Com isso, foi elaborada uma segunda versão do processo muito mais voltada para a realidade da empresa. Esta versão foi implantada nos demais projetos.

#### *2.5.1.3 Grupo de empresas*

Em Tsukumo *et al.* (2006), um grupo de empresas se uniram para definir e implantar seus processos. Inicialmente foi realizado um treinamento sobre introdução a melhoria de processos e aos modelos CMMI, MPS.BR e ISO/IEC 15504-5, focando nos modelos que seriam mais utilizados no programa de melhoria de processo. Além do treinamento, também se trabalhou na definição da estrutura dos documentos da empresa e na capacitação para definição do processo. Após, ocorreu um treinamento, o qual foi organizado por grupos de processo, que é um grupo com um ou mais processos. Durante o treinamento, o consultor deveria apoiar as definições de processo e sua implantação. Na implantação, ocorreu uma avaliação informal baseada no guia de avaliação do MPS.BR, o seu intuito foi permitir que a empresa percebesse os acertos obtidos até o momento e soubesse o que precisaria ser corrigido. Quando a empresa estivesse pronta seria realizada a avaliação formal.

#### *2.5.1.4 Empresa de Médio Porte*

Em Corgosinho (2006), foi utilizado o MPS.BR junto com a abordagem IDEAL (MCFEELY, 1996), que apresenta ações de iniciação, planejamento e execução de melhorias organizacionais, para definição e implantação de um processo de software em uma empresa de médio porte. As cinco fases do processo são explanadas no Quadro 5.

Quadro 5 – Fases do processo baseado na abordagem IDEAL

Fases	Descrição da fase
Iniciação	Nesta fase foi realizado um levantamento das expectativas sobre o programa de melhoria e o alinhamento dos projetos às estratégias da empresa.
Diagnóstico	Nesta fase foram levantados informações a respeito do estado atual da empresa e dos problemas enfrentados, realizado avaliação para verificar a maturidade da empresa e apresentados os resultados à alta gerência com o intuito de justificar a importância do programa de melhoria.
Planejamento	Os resultados da fase anterior serviram como entrada para esta, juntamente com a definição das prioridades, o planejamento das ações de melhoria, a aprovação do Plano do Programa de Melhoria com os envolvidos e a sua divulgação para toda a empresa. Além disso, para cada projeto, foi planejado atividades de levantamento e elaboração do processo correspondente.
Ação	Nesta fase o processo é definido, revisado e aprovado pela alta gerência, publicado para toda a empresa e apresentado em um treinamento de capacitação com todos os envolvidos. O processo foi aplicado em projetos pilotos dentro da empresa e auditorias semanais foram realizadas com o intuito de visualizar os pontos fortes e realizar ajustes necessários no processo.
Aprendizado e Ajustes	Nesta fase ocorreu uma reunião com os envolvidos para discutir as lições aprendidas em relação ao processo que foi seguido e os ajustes necessários que deveriam ser providenciados e acompanhados até a sua finalização. A nova versão do processo foi publicada e divulgada a todos os envolvidos e a partir dela, a alta gerência resolveu adotá-lo em todos os projetos.

Fonte: CORGOSINHO (2006).

### 2.5.2 Processos de Software no Ambiente Acadêmico

O processo de software traz diversos benefícios para as instituições que o adotam. Visando obter estes benefícios já constatados nas empresas, instituições de pesquisa, setores de TI dentro de universidades e órgãos do governo estão criando e implantando processos de software e se certificando em programas de melhoria de processo de software. Algumas destas instituições são o Centro de Computação da Aeronáutica de São José dos Campos (CCA SJ) (SCHEID *et al.*, 2007), o Laboratório de Engenharia de Software (LENS) da COPPE/UFRJ (SANTOS *et al.*, 2009), o Centro de Recursos Computacionais da Universidade Federal de Goiás (UFG) (MENDES *et al.*, 2010) e o Laboratório de Aplicações Ubíquas e Pervasivas (*Laboratory for Ubiquitous and Pervasive Applications - LUPA*) da UFG (MENDES; ALMEIDA; JUNIOR, 2011).

#### 2.5.2.1 Centro de Computação da Aeronáutica

O Centro de Computação da Aeronáutica de São José dos Campos (CCA SJ) é o órgão responsável pelo desenvolvimento, implantação, operação, manutenção e atualização dos sistemas operacionais de natureza e aplicação militar que apoiam as atividades da Força Aérea Brasileira.

Ele foi a primeira organização pública brasileira a ser avaliada no MR-MPS, no ano de 2006. Ele foi avaliado no nível E do MR-MPS. O Centro de Computação passou por

quatro experiências diferentes em melhoria de processos antes de iniciar a implantação de processos visando ao Nível E do MR-MPS. Ao iniciar a implementação, foram criados para cada processo do nível E um processo padrão adaptado as características do CCA SJ. Melhorias foram sugeridas e todas elas foram analisadas e registradas, as que não necessitavam que fosse gerada uma nova versão do processo foram institucionalizadas. Já as outras que necessitavam de modificação no processo foram implantadas posteriormente.

Durante a preparação para a avaliação, foram realizadas palestras e reuniões para explicar o processo e o método de avaliação do MPS.BR e o que era esperado da equipe do CCA SJ durante a avaliação. Ao fim da fase de reuniões, todos os envolvidos foram instigados a dar suas opiniões sobre a implantação dos processos. As respostas foram coletadas, analisadas e chegaram-se a alguns benefícios obtidos, os quais são explanados no Quadro 6 (SCHEID *et al.*, 2007, p. 5).

Quadro 6 – Benefícios obtidos com a implantação do processo do CCA SJ

<b>Benefícios obtidos com a implantação do processo</b>
Obteve-se um ganho de produtividade visível devido à diminuição do retrabalho.
Com a formalização dos papéis, a distribuição e alocação de tarefas tornou-se mais clara e permitiu-se saber a quem recorrer durante a realização de uma atividade.
Com a constante e frequente monitoração dos projetos melhorou a visibilidade do processo pela Alta Gerência e uma maior cobrança dos gerentes aos participantes do projeto. Além disso, o número de desvios no projeto foi reduzido drasticamente.
“A realização de reuniões periódicas para apresentar os relatórios de status dos processos melhorou a visibilidade das ações tomadas pelo grupo de processos e da aderência e adequação dos processos para a Alta Gerência”.
A formalização e obrigação de registro dos documentos foi importante por garantir uma boa documentação dos projetos.
Apesar de em alguns momentos ser percebido um aumento na burocracia, a eficiência no cumprimento das tarefas foi benéfica. Com um maior domínio do processo, a burocracia diminuiu com o tempo.
O comprometimento por parte da equipe permitiu a comunicação de indisponibilidade de realizar tarefas por falta de tempo ou por acúmulo de responsabilidades. Esta prática permitiu um melhor planejamento prévio por parte dos participantes do projeto.
Com a formalização do planejamento e da monitoração dos projetos, melhorou-se a previsão de entrega dos produtos.
Houve uma melhora na capacidade de organização.
Houve uma melhor divisão do tempo, tanto em relação às atividades a serem executadas quanto na alocação das pessoas nos diversos projetos.
Os projetos são gerenciados de acordo com as experiências passadas.

Fonte: SCHEID *et al.* (2007).

#### 2.5.2.2 Laboratório de Engenharia de Software da COPPE/UFRJ

Em 2008, a área de Qualidade de Software do Laboratório de Engenharia de Software (LENS) da COPPE/UFRJ foi a primeira unidade organizacional a ser avaliada no nível E do MR-MPS versão 1.2 (SANTOS *et al.*, 2009, p. 1).

Muitas das práticas previstas pelo modelo já eram utilizadas pela equipe e os envolvidos tinham experiência em implementação de processos, tanto com o modelo MPS.BR quanto com o modelo CMMI . Além disso, a estratégia de implementação da COPPE/UFRJ, a SPI-KM (SANTOS *et al.*, 2007), é baseada no uso de ferramentas, e já foi utilizada em vários casos de sucesso, incluindo organizações avaliadas no nível E do MR-MPS (por exemplo, (NUNES *et al.*, 2006; SCHEID *et al.*, 2007)) e nível 3 do CMMI (por exemplo, (FERREIRA *et al.*, 2007)). Alguns dos benefícios percebidos na implementação e implantação dos processos estão explanados no Quadro 7 (SANTOS *et al.*, 2009, p. 2).

Quadro 7 – Benefícios obtidos com a implantação do processo do LENS

<b>Benefícios obtidos com a implantação do processo</b>
Os requisitos mantiveram-se estáveis, resultado dos esforços para um correto e eficiente levantamento e verificação de requisitos.
Os procedimentos adotados nos testes foram tão eficientes que conseguiram capturar mais erros e retirá-los antes da avaliação externa.
As informações dos projetos são armazenadas e é gerada uma planilha para ser utilizada no planejamento dos projetos.
“Foi notado um grande número de melhorias no período inicial da implantação dos processos que diminuiu com o passar do tempo demonstrando a maior estabilidade e maturidade”.
Percebeu-se que as melhorias apresentadas a partir de um determinado momento implicavam em mudanças grandes nos processos e, por isso, foram programadas para serem implementadas em um novo ciclo de melhoria.
Percebeu-se que quanto maior o número de projetos e mais rápido eles evoluem, maior o número de reutilizações.
“Percebeu-se um grande pico na submissão e aprovação de novos itens de conhecimento decorrentes da institucionalização dos processos e da necessidade de se definir formas de operacionalizar e institucionalizar uma forma única de execução de certas atividades que eram descritas de forma pouco clara ou específica no processo de desenvolvimento de software”.

Fonte: SANTOS *et al.* (2009).

### 2.5.2.3 Centro de Recursos Computacionais da UFG

O Centro de Recursos Computacionais (CERCOMP) da Universidade Federal de Goiás (UFG) é o órgão responsável por gerenciar e manter a infraestrutura computacional, e projetar, desenvolver e manter sistemas computacionais da universidade. Ele possui três divisões – Sistemas, Redes e Suporte, sendo que a Divisão de Sistemas é a que realiza o desenvolvimento e a manutenção dos sistemas que atendem à UFG. Eles possuem atualmente um portfólio de 42 sistemas mantidos por uma equipe de 29 pessoas.

Em 2007, o CERCOMP iniciou um projeto de melhoria de processo de software. Foram realizados dois ciclos de melhoria e o planejamento do terceiro (MENDES *et al.*, 2010, p. 3).

No primeiro ciclo, foram executadas as fases previstas pelo método IDEAL: iniciação, diagnóstico, estabelecimento, ação e aprendizado. Na fase de iniciação, foi apresentado o roteiro geral de melhoria, as informações a respeito dos processos de software da Divisão de Sistemas foram coletadas e examinadas, e foi elaborado um plano preliminar do projeto. Na fase de diagnóstico, foi verificada a aderência dos processos de software da Divisão de Sistemas em relação o nível F do MPS.BR. O diagnóstico deu-se através de questionários online e entrevistas. Essas objetivaram esclarecer pontos obscuros nas respostas dos funcionários questionados. Ao final dessa fase, ocorreu uma reunião para apresentar e validar os resultados do diagnóstico com todos os membros da Divisão de Sistemas. Na fase seguinte, de estabelecimento, foi realizado um planejamento com base nos resultados do diagnóstico e estabelecidos quatro objetivos de melhoria. Na fase de ação, as soluções definidas para esses objetivos foram implementadas em um projeto piloto, o qual foi executado e acompanhado. Por fim, na fase de aprendizado, foi realizada uma reunião com o objetivo de apresentar os resultados alcançados, as lições aprendidas e indicar ações a serem realizadas no próximo ciclo de melhoria.

O segundo ciclo de melhoria teve como insumo os resultados alcançados no primeiro ciclo. A partir dessas informações percebeu-se a necessidade de implementar outras ações de melhoria. No final desse ciclo de melhoria, foi realizada uma pesquisa para avaliar a percepção dos colaboradores, coletar opiniões sobre o processo de software e identificar problemas e as dificuldades encontradas. Ao fim da pesquisa, foi elaborado um relatório o qual continha os resultados obtidos. Esses foram apresentados para todos os funcionários da Divisão de Sistemas durante uma reunião e utilizados para planejar o terceiro ciclo de melhoria.

#### *2.5.2.4 Laboratory for Ubiquitous and Pervasive Applications (LUPA) da UFG*

O Laboratório de Aplicações Ubíquas e Pervasivas (*Laboratory for Ubiquitous and Pervasive Applications - LUPA*) é um laboratório de pesquisa da UFG que desenvolve soluções de geotecnologia através de interfaces ricas para a web e computação distribuída e paralela (MENDES; ALMEIDA; JUNIOR, 2011).

Em 2007, o LUPA iniciou um programa de melhoria de processos. Para a implantação do processo foi escolhido um processo semelhante ao IDEAL (MCFEELEY, 1996). As seis fases do processo do LUPA são explanadas no Quadro 8.

Quadro 8 – Fases do processo do LUPA

Fases	Descrição da fase
Iniciação	Nesta fase foi realizada uma entrevista não estruturada com o coordenador do laboratório de pesquisa, com o intuito de identificar os pontos fracos do modo de trabalho e assim captar as principais necessidades do processo a ser construído. Porém, ainda era necessária conhecer um pouco mais o modo de trabalho.
Diagnóstico	Nela foi realizada uma entrevista conjunta com o coordenador do grupo de pesquisa e com um desenvolvedor. O diagnóstico teve como base o nível G do MPS.BR. Cada resultado esperado deste nível foi transformado em um tópico da entrevista. Ao fim desta, foi gerado um relatório de diagnóstico o qual continha as principais não conformidades com o MPS.BR. Além disso, também foram detectados outros pontos fracos e a necessidade de abranger outros processos além daqueles do nível G.
Planejamento	Nesta fase os processos foram selecionados e priorizados para treinamento, e foi definida a ferramenta que seria utilizada na definição do processo.
Definição dos processos	Para definir o processo optou-se por utilizar mais de um modelo de melhoria de processo e também duas metodologias ágeis. Além disso, percebeu-se a necessidade de incluir outras áreas de melhoria.
Treinamento	No treinamento foram explanados conceitos fundamentais e apresentado o fluxograma e detalhes das atividades das áreas do processo priorizados na fase de planejamento.
Execução do processo	Durante a execução houve uma orientação sobre o que era esperado de cada processo e como deveria ser preenchido cada <i>template</i> . O grupo de processo investiu na construção de guias de execução do processo e exemplos de cada resultado. Esses guias eram informais e continham informações a respeito das atividades e dos <i>templates</i> . Os guias foram validados com os líderes dos projetos e refeitos caso não atingissem seu objetivo. Após os ajustes, ocorreu uma avaliação de conformidade da execução com a definição, realizada pelo grupo de processos, a qual serviu como consultoria para o grupo.

Fonte: MENDES, ALMEIDA, JUNIOR (2011).

## 2.6 Conclusão da Seção

Nesta Seção foi apresentado a fundamentação teórica desse trabalho, explanando todos os conceitos-chave para o bom entendimento deste. Esta Seção apresentou um breve resumo sobre processo de desenvolvimento de software, algumas definições de processo de software, uma breve descrição dos modelos de melhoria de processo, guia e metodologias de desenvolvimento de software utilizados, e alguns trabalhos de definição e implantação de processo de software no ambiente empresarial e acadêmico. Nestes trabalhos podemos perceber que já existem diversos processos definidos e implantados no ambiente acadêmico, porém apesar dos esforços, durante a construção deste trabalho não foram localizadas iniciativas de implantação de processos de software no contexto de Núcleo de Práticas de uma universidade, por isso este trabalho objetivou atuar nesse sentido.

A Seção seguinte apresenta os procedimentos metodológicos utilizados na construção deste trabalho.

### 3 PROCEDIMENTOS METODOLÓGICOS

Este trabalho visa à definição e implantação de um processo de software no Núcleo de Práticas de Informática de uma universidade federal. Para alcançar este objetivo, algumas etapas foram estimadas. As etapas para a construção deste trabalho são:

- **primeira etapa:** Capturar e analisar o desenvolvimento das atividades no NPI;
- **segunda etapa:** Realizar pesquisa de trabalhos de definição e implantação de processos em ambiente empresarial e acadêmico, e selecionar melhores práticas de modelos de processos de software tradicionais e ágeis;
- **terceira etapa:** Definição de um processo baseado nas boas práticas dos modelos e metodologias estudados, e na análise das atividades desenvolvidas no NPI;
- **quarta etapa:** Implantação do processo em projetos pilotos do NPI e coleta dos resultados.

As quatro etapas deste trabalho são descritas a seguir.

A primeira etapa consiste na captura do desenvolvimento das atividades do NPI. Esta captura se dará por meio de entrevistas e reuniões com os líderes técnicos dos projetos e com o professor supervisor. Após a captura dos dados, esses serão analisados e servirão de entrada para a definição do processo. A análise consiste da extração de informações a respeito das atividades desenvolvidas, ferramentas utilizadas, artefatos gerados, entre outros aspectos.

A segunda etapa consiste na seleção de melhores práticas de modelos de processos de software tradicionais e ágeis, focando nas disciplinas de Gerência de Projetos, Requisitos e Gerência de Configuração. A seleção será realizada baseada nas necessidades do NPI, que serão identificadas em reuniões com o professor supervisor. Também nesta etapa serão pesquisados trabalhos que definiram e implantaram processos de software em ambiente empresarial e acadêmico.

A terceira etapa visa à definição do processo. O processo será definido baseado na análise do desenvolvimento das atividades do Núcleo de Práticas, na seleção das boas práticas dos modelos de processos de software tradicionais e ágeis. O processo será modelado na ferramenta *Eclipse Process Framework Composer* (EPF Composer) (ECLIPSE FOUNDATION, 2011). Ela foi escolhida, pois é bastante utilizada por empresas de desenvolvimento de software para modelar processos.

A quarta fase consiste na implantação do processo. O processo será implantado em projetos pilotos do NPI. Ao início da implantação acontecerá um treinamento para explanação do processo a todos os integrantes dos projetos. Com o intuito de garantir que o processo proposto seja seguido, será realizado um acompanhamento da utilização deste processo. Ao final dos projetos deve-se gerar um relatório analisando a implantação com os resultados.

### **3.1 Conclusão da Seção**

Esta Seção apresentou a metodologia utilizada para a execução deste trabalho. A Seção seguinte apresenta todas as fases definidas para a construção deste trabalho.

## **4 DEFINIÇÃO DO PROCESSO DO NPI**

Esta Seção tem como objetivo apresentar todas as etapas de elaboração deste trabalho. Esta Seção é composta de cinco subseções. A subseção 4.1 apresenta a primeira etapa, na qual foi realizado uma captura das atividades de desenvolvimento do NPI e sua análise. A subseção 4.2 apresenta a segunda etapa, na qual foi realizada uma pesquisa sobre trabalhos de definição e implantação de processo e a seleção de boas práticas de modelos de processo de software. A subseção 4.3 apresenta a definição do processo propriamente dita, na qual as informações das duas etapas anteriores serviram como insumo e diversas versões do processo foram definidas até se chegar a uma versão final para implantação em projetos pilotos. A subseção 4.4 apresenta a implantação do processo. A subseção 4.5 apresenta a conclusão desta Seção.

### **4.1 Primeira etapa**

A primeira etapa teve início com reuniões com o professor supervisor do NPI e outros professores. Estas reuniões tinham como intuito capturar a forma como as atividades são desenvolvidas no NPI, mas apesar do conteúdo das reuniões ter servido de entrada para o desenvolvimento de versões iniciais do processo, era necessária uma formalização.

Com o intuito de formalizar a maneira como as atividades são desenvolvidas no NPI, foi realizada uma entrevista não-estruturada com o professor supervisor e os líderes técnicos dos projetos que estavam sendo desenvolvidos no NPI. Os líderes técnicos são os responsáveis por coordenar as equipes dos projetos, onde cada projeto está associado a um líder técnico. A entrevista foi aplicada com quatro integrantes do NPI, de maneira individual e objetivou capturar as atividades realizadas nos projetos, como fases do projeto, atividades realizadas dentro das fases, artefatos gerados, qual metodologia de engenharia de software é utilizada, entre outros aspectos. A entrevista serviu como uma captura inicial do desenvolvimento das atividades do NPI.

Na entrevista foram feitas doze perguntas a cada integrante do NPI. As perguntas e as respostas consolidadas serão explanadas no Quadro 9.

Quadro 9 – Perguntas e respostas da entrevista

Perguntas	Respostas consolidadas
O Núcleo de Práticas já possui algum processo definido? Se sim, defina-o.	Dois integrantes responderam que não e os outros disseram que sim. Os que responderam sim, relataram um pouco do processo de desenvolvimento de software do NPI e um deles afirmou que o processo não estava documentado, porém teria sido explanado aos integrantes dos projetos, bem no início das atividades.
São utilizadas metodologias, guias, modelos de melhoria de processo para melhorar a execução dos projetos?	Todos disseram que os projetos utilizavam Scrum e apenas um entrevistado respondeu XP ( <i>Extreme Programming</i> ), mas não relatou quais práticas dessa metodologia eram utilizadas. No Scrum, um entrevistado citou algumas práticas, <i>Sprint</i> , Reunião de Planejamento da <i>Sprint</i> e Reunião Diária.
Como os projetos chegam e são selecionados (Por demanda, interna ou externa, ou outros)	Dois entrevistados responderam que o professor supervisor que seleciona os projetos. Os outros dois explanaram que alguns projetos partem de convênios com a universidade e por demanda interna. O professor supervisor entra em contatos com os parceiros e clientes em potencial para conseguir os projetos.
Como são alocadas as equipes para os projetos selecionados? Quem coordena as equipes?	Cada equipe é coordenada por um líder técnico. Eles são escolhidos pelo professor supervisor. Os demais alunos tem o livre arbítrio de escolher a sua equipe. A escolha geralmente é feita por compatibilidade de horários e por afinidade, nessa ordem de prioridade.
Qual a média de pessoas alocadas para os projetos	Todos responderam que a média de pessoas nos projetos era entre 3 a 5 membros.
Qual o primeiro passo a ser feito no início de qualquer projeto?	As respostas foram diversas, pois cada um relatou o seu projeto. Foi citado a escolha dos integrantes do projeto, os membros do projeto recebem os requisitos se já houver e iniciam a sua elicitação, membros mantem compromisso com o projeto, definem tecnologia a ser utilizada no desenvolvimento do produto, definem os primeiros casos de uso a serem desenvolvidos, estudam novas tecnologias que serão utilizadas no projeto e se reúnem com o cliente.
Quais os papéis e as responsabilidades dos integrantes da equipe.	Todas as respostas foram semelhantes. Líder técnico – Coordena e orienta a equipe e também desenvolve as funções de um programador. Programador – Elicita os requisitos, desenvolve e testa o sistema, entrevista o cliente e realiza capacitação. Professor supervisor – Capta, gerencia e acompanha as atividades do projeto.
O projeto possui fases? Quais são? Descreva como ocorre essas fases.	O projeto pode ser novo ou pode ser continuado em outro semestre. Inicialmente os requisitos de cada projeto são elicitados e escolhidos para serem desenvolvidos. Caso o projeto já tenha requisitos elicitados, eles são escolhidos e implementados. Os requisitos são implementados, testados individualmente e depois integrados e o produto da sua integração é testado. Os erros encontrados devem ser corrigidos e inseridos na planilha de teste. O encerramento do projeto acontece e caso o projeto tenha sido concluído, o produto final deve ser implantado no ambiente escolhido pelo cliente.
Como ocorrem as entregas dos projetos e como estes são validados pelo cliente?	Ocorrem entregas parciais, onde o professor supervisor testa o que foi realizado durante a <i>Sprint</i> e os erros encontrados devem ser corrigidos até o final da <i>Sprint</i> seguinte. Somente o produto final é entregue ao cliente.
Como os alunos são avaliados?	Os alunos são avaliados pelas entregas parciais e pelo produto final. Um entrevistado disse que os líderes técnicos são avaliados mensalmente por meio de relatório, contendo todas as atividades desenvolvidas por ele e os demais alunos pelas entregas parciais.
Se existem, explique os artefatos desenvolvidos durante cada fase do projeto.	Todos os entrevistado somente citaram os artefatos utilizados sem relacioná-los as fases do projeto. Os artefatos citados foram Documento de Requisitos, Documento de casos de uso, Relatório inicial e final do estágio, código, Planilha de testes (geralmente não utilizada nos projetos) e manual do <i>Entities</i> .
Algum tipo de ferramenta é utilizado? Quais?	Diversas ferramentas foram citadas. Elas são <i>NetBeans</i> , <i>Entities</i> , <i>PostgreSQL</i> , <i>Astah</i> (modelagem UML), <i>Tomcat</i> , <i>Word</i> , <i>Excel</i> , <i>Browser</i> .
São realizados workshops para aperfeiçoar os conhecimentos da equipe?	Sim. Ocorreram dois eventos de aperfeiçoamento, um minicurso durante a Escola de Verão sobre o <i>Entities</i> , recomendado a todos os integrantes do NPI, mas não obrigatório e uma palestra avançada de <i>Entities</i> durante o WTISC, obrigatória. Além disso, houve uma capacitação para geração de relatório no <i>Entities</i> .

Fonte: a própria autora.

As respostas obtidas com a entrevista demonstraram que o NPI não possui um processo definido, porém existe um fluxo de trabalho definido, mas este não está documentado; os projetos têm pouca ou nenhuma documentação; as únicas ferramentas utilizadas nos projetos são voltadas somente para o desenvolvimento de software; existem poucas atividades relacionadas ao gerenciamento de projeto; são utilizadas metodologias ágeis para o desenvolvimento, porém são usados somente alguns conceitos destas e elas não são aplicadas completamente.

O conhecimento obtido a partir das respostas juntamente com as informações coletadas durante as reuniões com os envolvidos no NPI serviu para conhecer quais as fases do projeto, quais as suas atividades, quem são os responsáveis pelas atividades, quais artefatos servem de entrada e saída, quais os papéis e as responsabilidades dos integrantes do NPI, entre outros. Todas essas informações serviram de entrada para a construção do processo de software do NPI.

## **4.2 Segunda etapa**

Na segunda etapa deste trabalho foi realizada uma pesquisa de trabalhos de definição e implantação de processos no ambiente empresarial e acadêmico, com o intuito de aprender quais modelos, técnicas, ferramentas apoiaram a construção do trabalho e conhecer as dificuldades enfrentadas por eles. Nestes trabalhos percebemos alguns passos que deveriam ser utilizados para se chegar a definição do processo e também sua implantação, como:

- identificar situação atual da organização. O status atual pode ser obtido por meio de observações, questionários e entrevistas;
- identificar os pontos fracos no modo de trabalho e assim captar as principais necessidades do processo a ser construído;
- realizar Plano de Melhoria com todas as melhorias propostas para serem implantadas no processo;
- padronização dos documentos;
- realizar uma capacitação do processo proposto;
- execução do processo em projetos pilotos;
- avaliação da aderência do processo;
- realizar reunião com os envolvidos para discutir as lições aprendidas;

- acompanhar a implantação do processo no projeto piloto e documentar as dificuldades encontradas para posterior análise e avaliação dos processos definidos.

Neste trabalho todos esses passos foram utilizados para a definição e implantação do processo.

Também nesta etapa, foi realizado um levantamento dos modelos de processos de software tradicionais e ágeis (CMMI, MPS.BR, PMBoK, RUP, Scrum e SCORE). Nesses modelos foram observadas práticas interessantes para serem incorporadas no processo. Foram selecionadas algumas práticas focando nas áreas de Gerência de Configuração e Requisitos, porém outras práticas além dessas áreas também foram selecionadas.

O Quadro 10 apresenta as práticas selecionadas, associadas ao seu respectivo modelo e a tarefa no processo onde esta prática foi aplicada.

Quadro 10 – Mapeamento das boas práticas dos modelos e o processo do NPI

Modelo	Prática do modelo	Tarefa do Processo
PMBok	Escopo – Coletar os requisitos	Elicitar requisitos
MPS.BR	GRE1	
CMMI	DR – SP 1.1 Levantar Necessidades	
MPS.BR	GRE1	Validar requisitos
Scrum	Documento: <i>Backlog</i> do Produto	Criar <i>Backlog</i> do Produto
Scrum	Reunião de Planejamento da <i>Sprint</i>	Realizar Reunião de Planejamento da <i>Sprint</i>
RUP	Requisitos – Detalhar um caso de uso	Especificar Caso de Uso
	Requisitos – Localizar atores e casos de uso	
RUP	Ambiente – Desenvolver Guia de Ferramentas	Preparar ambiente
RUP	Implementação – Integrar subsistema	Gerar <i>release</i>
	Implementação – Integrar sistema	
Scrum	Revisão da <i>Sprint</i>	Realizar Reunião de Revisão da <i>Sprint</i>
Scrum	Reunião Diária	Realizar Reunião Diária
SCORE	Reunião de <i>Status</i>	Realizar Reunião de <i>Status</i> do NPI
CMMI	CM – SP 1.1 Identificar Itens de Configuração	Criar Plano de Gerenciamento de Configuração
MPS.BR	GCO 2	
CMMI	CM – SP 1.2 Estabelecer um Sistema de Gestão de Configuração	Estabelecer um Sistema de Gestão de Configuração
MPS.BR	GCO1	
RUP	Preparar Finalização do Projeto	Realizar Reunião de Encerramento do Projeto
PMBok	Integração - Encerrar o projeto ou fase	

Fonte: a própria autora.

O Quadro 11 apresenta alguns documentos do processo, os artefatos de onde eles foram adaptados, associados ao seu respectivo modelo.

Quadro 11 - Mapeamento do artefatos dos modelos e o processo do NPI

<b>Modelo</b>	<b>Artefato do modelo</b>	<b>Documento do Processo</b>
RUP	Requisitos – Visão	Documento de Visão
RUP	Requisitos – Especificação de Requisitos de Software	Documento de Especificação dos Requisitos
Scrum	Incremento	<i>Release</i> do Produto

Fonte: a própria autora.

### 4.3 Terceira etapa

A terceira etapa teve como entrada o conhecimento obtido na primeira etapa, através do estudo dos modelos e as informações coletadas sobre o processo de desenvolvimento de software do NPI na segunda etapa.

A definição do processo passou por diversas versões. Inicialmente foi definida a primeira versão do processo baseada em um processo que estava sendo desenvolvido por um bolsista do NPI, porém este processo estava muito extenso e a maioria das suas atividades não estava de acordo com as atividades reais que o NPI desenvolvia. A primeira versão do processo deveria ser algo simples e que mostrasse a realidade das atividades desenvolvidas no NPI, ou seja, um autorretrato do estado atual do Núcleo.

A partir dessa primeira versão, outras versões foram desenvolvidas baseadas no conhecimento adquirido em reuniões com o professor supervisor e outros professores. Nessas reuniões, os professores repassaram a forma como as atividades no NPI eram desenvolvidas e sugeriam melhorias no processo. Após cada reunião, foi gerada uma nova versão do processo com as mudanças sugeridas. Além do conhecimento agregado a partir das reuniões, foram selecionadas algumas práticas dos modelos de melhoria de processo estudados, do guia de conhecimento em gerência de projetos e de metodologias tradicional e ágil. Essas práticas foram inseridas no processo. Após diversas versões, chegou-se a uma versão final do processo. Esta versão será explanada na próxima seção.

### 4.4 Quarta etapa

A quarta etapa consistiu da implantação da versão final do processo em projetos pilotos do NPI.

A implantação do processo de software do NPI começou em abril de 2013, junto com as atividades da universidade. O NPI, a partir desta data, se dividiu em dois grupos, o grupo dos estudantes do curso de Sistemas de Informação e do curso de Engenharia de

Software. O processo foi idealizado e definido para ser implantado no grupo de Sistemas de Informação, porém por conta dos projetos que iriam ser implementados naquele semestre não estarem dentro do escopo do processo, optou-se por implantá-lo no grupo de Engenharia de Software.

Inicialmente, foi realizado um treinamento do processo a todos os integrantes dos dois grupos do NPI, detalhando todas as atividades do processo, suas descrições, seus passos e os artefatos que nela deveriam ser gerados. Todos os artefatos anexados ao processo foram expostos e descritos detalhadamente.

Após o treinamento, deu-se início as atividades do NPI e a implantação do processo. Este foi implantado em dois projetos pilotos dentro do grupo de Engenharia de Software. Um projeto iniciou com sete integrantes e no meio do projeto estava com quatro, e o outro iniciou com 3 integrantes e no meio do projeto mais dois integrantes foram alocados. O acompanhamento do processo foi realizado pelo bolsista de processo do NPI.

Enquanto o projeto estava em execução, um *checklist* foi desenvolvido para medir a aderência da execução com o processo proposto. O *checklist* tem como intuito revisar as atividades e identificar se elas foram executadas de acordo com o processo. O *checklist* é composto de perguntas sobre cada tarefa do processo.

Os projetos, até o fechamento deste trabalho, somente tinham executado as atividades de Iniciar Projeto, Requisitos, Gerenciamento do Projeto e Gerenciamento de Configuração. As demais atividades ainda não tinham sido iniciadas, por isso o *checklist* não pode ser aplicado por completo. O *checklist* completo está disponível no Apêndice A.

Além do *checklist*, foi realizada uma entrevista com os líderes técnicos e com o professor supervisor para entender como ocorreu a execução do processo, as não conformidades que existiram, sugestão de melhoria e a visão deles sobre o processo. As respostas do *checklist*, bem como a visão dos entrevistados sobre o processo, as dificuldades, as lições aprendidas e oportunidades de melhoria identificadas são apresentadas na seção 6, Implantação do Processo no NPI.

#### **4.5 Conclusão da Seção**

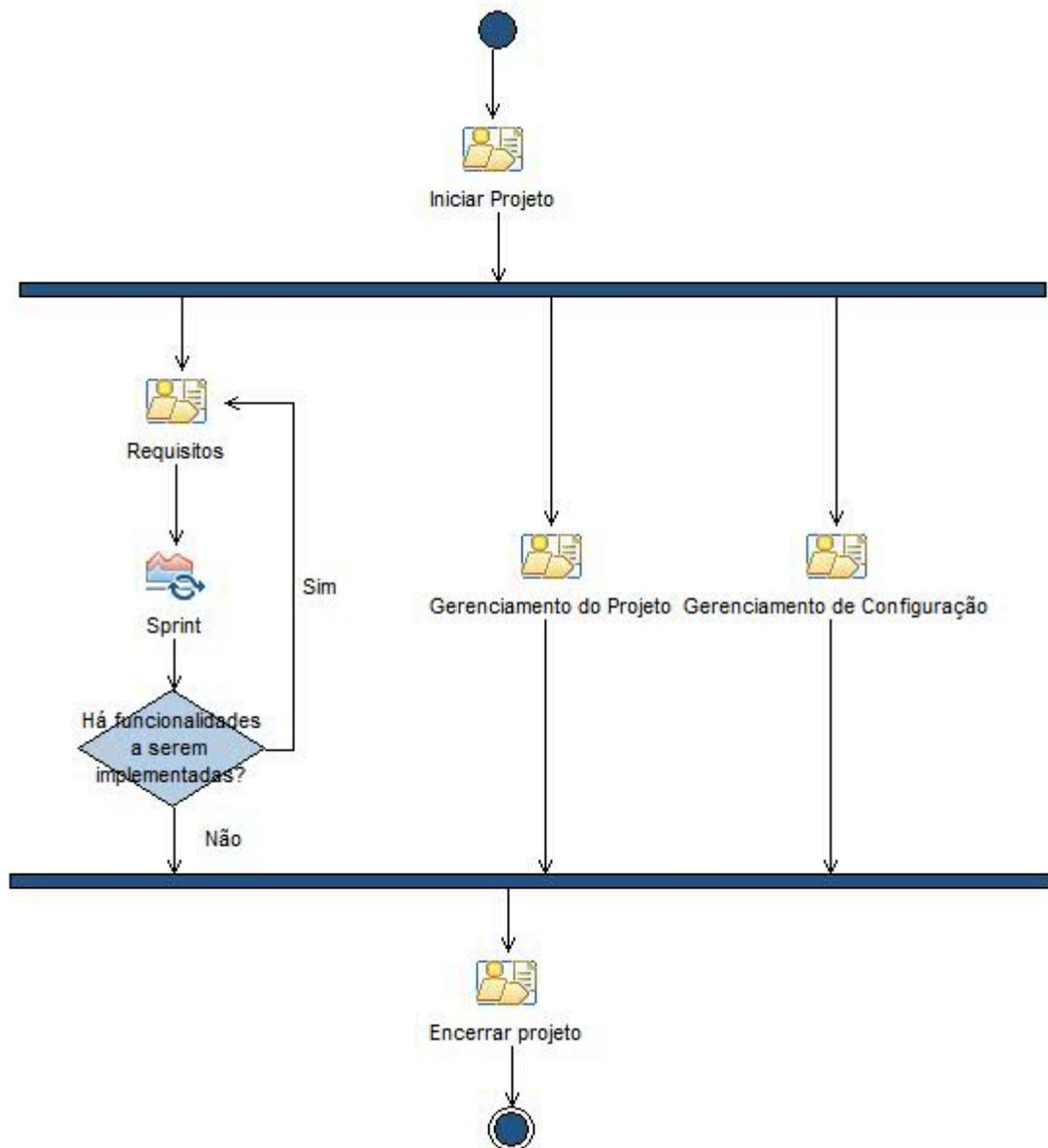
Esta Seção apresentou detalhadamente todas as etapas de execução deste trabalho. A próxima Seção apresenta detalhadamente todas as atividades e tarefas do processo proposto.

## 5 PROCESSO DO NPI

Esta Seção tem como objetivo apresentar o processo proposto, o qual foi implantado em projetos pilotos no NPI.

O processo de software desenvolvido para o NPI foi modelado na ferramenta EPF Composer. Ele descreve as atividades e os passos necessários para a sua execução, com o intuito de guiar o desenvolvimento de software do NPI. A seguir, a Figura 2 apresenta o fluxo principal do processo com todas as suas atividades e a sua única iteração.

Figura 2 – Fluxo principal do processo de software do Núcleo de Práticas de Informática



Fonte: a própria autora.

A seguir, são apresentadas detalhadamente todas as atividades do processo proposto.

### 5.1 Atividade “Iniciar projeto”

A atividade “Iniciar Projeto” é a primeira a ser executada. Ela é uma atividade de iniciação dos projetos. Dentro desta atividade existem algumas tarefas que devem ser realizadas e estas devem ser executadas segundo um fluxo. A Figura 3 apresenta o fluxo da atividade “Iniciar Projeto” e suas tarefas.

Figura 3 – Fluxo da atividade “Iniciar Projeto”



Fonte: a própria autora.

#### 5.1.1 Tarefa “Prospectar Projetos”

A tarefa “Prospectar Projetos” objetiva a captura dos futuros projetos a serem desenvolvidos pelo NPI. Para a execução desta tarefa algumas informações precisam ser conhecidas. O Quadro 12 apresenta as informações referentes à tarefa “Prospectar Projetos”.

Quadro 12 – Representação da tarefa “Prospectar Projetos”

<b>Objetivo:</b> Esta tarefa tem como objetivo prospectar projetos para serem desenvolvidos no NPI.	
<b>Papéis principais:</b> Cliente Professor supervisor	<b>Papéis secundários:</b> -
<b>Entrada:</b> -	<b>Saída:</b> Documento de Visão
<b>Passos:</b> 1) Projetos devem ser prospectados	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Projetos devem ser prospectados:** O professor supervisor é o responsável pelo Núcleo de Práticas de Informática (NPI) e este deverá prospectar projetos. O professor supervisor terá encontro com possíveis fornecedores de projetos e será responsável por captar projetos. Ao ser fechado um projeto, o professor supervisor deverá criar um documento de visão com todos os detalhes conseguidos durante as reuniões com o fornecedor.

### 5.1.2 Tarefa “Realizar Reunião de Kickoff”

A tarefa “Realizar Reunião de *Kickoff*” objetiva realizar uma reunião onde deve ser explanado os projetos a todos os integrantes do NPI e os recursos de cada projeto devem ser alocados. O Quadro 13 apresenta as informações referentes à tarefa “Realizar Reunião de *Kickoff*”.

Quadro 13 – Representação da tarefa “Realizar Reunião de *Kickoff*”

<b>Objetivo:</b> Esta tarefa tem como objetivo realizar uma reunião de <i>kickoff</i> com o intuito de apresentar os projetos aos integrantes do NPI, bem como alocar os recursos necessários para cada projeto e escolher os líderes técnicos.	
<b>Papéis principais:</b> Professor supervisor Equipe de desenvolvimento	<b>Papéis secundários:</b> -
<b>Entrada:</b> Documento de Visão	<b>Saída:</b> Ata de Reunião
<b>Passos:</b> 1) Realizar reunião de <i>kickoff</i>	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Realizar reunião de *kickoff*:** A Reunião de *Kickoff* tem como intuito apresentar os projetos a todos os envolvidos do NPI. O responsável por essa reunião e o seu conteúdo é o professor supervisor. Nessa reunião devem ser escolhidos os líderes técnicos e

alocados os recursos que irão trabalhar em cada projeto e determinado quais suas responsabilidades. Nesta atividade todos os participantes terão que se comprometer com o projeto. Deve ser criada uma ata de reunião documentando tudo o que foi dito durante a Reunião de *Kickoff*.

### 5.1.3 Tarefa “Elaborar Relatório Inicial”

A tarefa “Elaborar Relatório Inicial” objetiva a construção do relatório inicial do estágio. O Quadro 14 apresenta as informações referentes à tarefa “Elaborar Relatório Inicial”.

Quadro 14 – Representação da tarefa “Elaborar Relatório Inicial”

<b>Objetivo:</b> Esta tarefa tem como objetivo elaborar o relatório inicial do estágio.	
<b>Papéis principais:</b> Equipe de desenvolvimento	<b>Papéis secundários:</b> -
<b>Entrada:</b> -	<b>Saída:</b> Relatório Inicial
<b>Passos:</b> 1) Elaborar relatório inicial	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Elaborar relatório inicial:** Deve ser elaborado um relatório inicial o qual deve conter: os dados dos integrantes do NPI; os dados da instituição do estágio; uma pequena introdução descrevendo a empresa onde será realizado o estágio e informações gerais sobre as atividades que serão desenvolvidas; descrever as qualificações do estagiário; citar em tópicos gerais os objetivos do estágio; justificar a importância do local de trabalho e das atividades a serem realizadas para a carreira profissional; desenvolver um cronograma das atividades a serem realizadas. O *template* do relatório final deve servir de guia para a construção do relatório.

## 5.2 Atividade “Requisitos”

A atividade “Requisitos” é responsável pela elicitação e validação dos requisitos do projeto. A Figura 4 apresenta o fluxo da atividade “Requisitos” e suas tarefas.

Figura 4 – Fluxo da atividade “Requisitos”



Fonte: a própria autora.

### 5.2.1 Tarefa “Elicitar Requisitos”

A tarefa “Elicitar Requisitos” objetiva identificar e especificar de maneira formal os requisitos do software a ser construído. O Quadro 15 apresenta as informações referentes à tarefa “Elicitar Requisitos”.

Quadro 15 – Representação da tarefa “Elicitar Requisitos”

<b>Objetivo:</b> Esta tarefa consiste na identificação e especificação de maneira formal dos requisitos do software a ser construído.	
<b>Papéis principais:</b> Equipe de desenvolvimento Líder técnico	<b>Papéis secundários:</b> Professor supervisor
<b>Entrada:</b> Documento de Visão	<b>Saída:</b> Documento de Especificação dos Requisitos
<b>Passos:</b> 1) Identificar os requisitos 2) Especificar os requisitos	

Fonte: a própria autora.

A seguir é apresentada a descrição dos passos desta tarefa.

**Passo 1 – Identificar os requisitos:** O líder técnico, juntamente com a equipe do projeto terá como base o documento de visão para começar o projeto, porém este documento pode não ser suficiente. Portanto, a equipe poderá realizar reuniões com o cliente

e através de entrevistas, observação direta ou outras técnicas de elicitación, será feito o levantamento das necessidades do sistema. Essas necessidades serão documentadas através de texto escrito ou de gravações das entrevistas.

**Passo 2 – Especificar os requisitos:** O documento especificação dos requisitos deve ser criado com base na documentação criada no passo anterior. Nele deve conter uma breve descrição do documento e todos os requisitos identificados no passo anterior.

### 5.2.2 Tarefa “Validar Requisitos”

A tarefa “Validar Requisitos” objetiva a validação dos requisitos pelo cliente. O Quadro 16 apresenta as informações referentes à tarefa “Validar Requisitos”.

Quadro 16 – Representação da tarefa “Validar Requisitos”

<b>Objetivo:</b> Esta tarefa consiste na validação dos requisitos pelo cliente.	
<b>Papéis principais:</b> Cliente Equipe de desenvolvimento Líder técnico	<b>Papéis secundários:</b> -
<b>Entrada:</b> Documento de Especificação dos Requisitos	<b>Saída:</b> Documento de Especificação dos Requisitos
<b>Passos:</b> 1) Validar requisitos pelo cliente 2) Realizar alterações nos requisitos	

Fonte: a própria autora.

A seguir é apresentada a descrição dos passos desta tarefa.

**Passo 1 – Validar os requisitos pelo cliente:** Os requisitos identificados, os quais estão contidos no documento de especificação dos requisitos, devem ser validados pelo cliente. O cliente tem o poder de adicionar, modificar e excluir requisitos.

**Passo 2 – Realizar alterações nos requisitos:** Caso seja realizado quaisquer alterações nos requisitos, estas devem ser registradas no documento de especificação de requisitos.

### 5.2.3 Tarefa “Criar Backlog do Produto”

A tarefa “Criar *Backlog* do Produto” objetiva a construção do documento *Backlog* do Produto, o qual deverá conter todos os requisitos elicitados e validados. O Quadro 17 apresenta as informações referentes à tarefa “Criar *Backlog* do Produto”.

Quadro 17 – Representação da tarefa “Criar *Backlog* do Produto”

<b>Objetivo:</b> Esta tarefa consiste na construção do <i>Backlog</i> do Produto, que é um arquivo onde devem estar contidos todos os requisitos revisados pela equipe.	
<b>Papéis principais:</b> Equipe de desenvolvimento Líder técnico	<b>Papéis secundários:</b> -
<b>Entrada:</b> Documento de Especificação dos Requisitos	<b>Saída:</b> <i>Backlog</i> do Produto
<b>Passos:</b> 1) Construção do <i>backlog</i> do produto	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Construção do *backlog* do produto:** O *Backlog* do Produto deve ser criado a partir do documento de especificação dos requisitos. Todos os requisitos contidos neste documento devem ser inseridos no *Backlog*. O *Backlog* do Produto é composto de três planilhas. Nesta tarefa a única planilha que deve ser preenchida é a que tem o nome *Backlog* do Projeto. Nessa tarefa todos os requisitos devem ser inseridos na planilha e divididos por tema (por exemplo, *login*, CRUD Cadastro), deve ser dado um número identificador único, um nome, uma descrição, um tipo (por exemplo, requisito funcional, requisito não funcional) para cada requisito, além disso, deve ser dado um status para o requisito como criado, estimado, planejado ou concluído. Esse status descreve o estado do requisito dentro do projeto, ou seja, se ele foi criado, ele somente se encontra inserido nesta planilha, se ele foi planejado, significa que ele foi planejado para ser construído em uma *Sprint*, se seu status é concluído, significa que o requisito já foi construído.

## 5.3 Iteração “Sprint”

Uma iteração é uma fase a qual pode ser repetida diversas vezes. A iteração “*Sprint*” é um contêiner para diversas atividades. Dentro do projeto deverão ocorrer mais de uma *Sprint*. Enquanto tiverem requisitos a serem desenvolvidos haverá uma *Sprint* em

execução e quando essa iteração ocorre, deve-se passar por todas as atividades que estão contidas dentro dela. A Figura 5 apresenta o fluxo da iteração “*Sprint*” e suas atividades.

Figura 5 – Fluxo da iteração “*Sprint*”



Fonte: a própria autora.

### 5.3.1 Atividade “*Planejar Sprint*”

A atividade “*Planejar Sprint*” objetiva realizar o planejamento do que será desenvolvido durante a *Sprint*. A Figura 6 apresenta o fluxo da atividade “*Planejar Sprint*” e sua única tarefa.

Figura 6 – Fluxo da atividade “Planejar *Sprint*”

Fonte: a própria autora.

#### 5.3.1.1 Tarefa “Realizar Reunião de Planejamento da *Sprint*”

A tarefa “Realizar Reunião de Planejamento da *Sprint*” objetiva planejar o que será desenvolvido durante a *Sprint* atual. O Quadro 18 apresenta as informações referentes à tarefa “Realizar Reunião de Planejamento da *Sprint*”.

Quadro 18 – Representação da tarefa “Realizar Reunião de Planejamento da *Sprint*”

<b>Objetivo:</b> Esta tarefa tem como objetivo realizar a Reunião de Planejamento da <i>Sprint</i> , onde deve ser decidido tudo o que será desenvolvido na <i>Sprint</i> atual.	
<b>Papéis principais:</b> Equipe de desenvolvimento Líder técnico	<b>Papéis secundários:</b> -
<b>Entrada:</b> <i>Backlog</i> do Produto	<b>Saída:</b> <i>Backlog</i> do Produto
<b>Passos:</b> 1) Realizar reunião de planejamento da <i>Sprint</i> 2) Selecionar itens no <i>backlog</i> do produto	

Fonte: a própria autora.

A seguir é apresentada a descrição dos passos desta tarefa.

**Passo 1 – Realizar reunião de planejamento da *Sprint*:** O planejamento da *Sprint* dá se através de uma reunião de planejamento com a presença do líder técnico e da equipe de desenvolvimento para selecionar os casos de uso que serão desenvolvidos na *Sprint* atual. Cada *Sprint* terá duração de 15 dias.

**Passo 2 – Selecionar itens no *backlog* do produto:** O *Backlog* do Produto é uma lista de trabalhos a serem concluídos pelo time. O líder técnico junto com a equipe de desenvolvimento irá selecionar itens (requisitos) do *Backlog* do Produto que serão desenvolvidos na *Sprint* atual. Os itens selecionados nas primeiras *Sprints* devem ser os

requisitos mais simples e que não dependam de outros requisitos. Os requisitos devem ser priorizados, ou seja, primeiro devem ser selecionados os requisitos com maior valor a ser agregado ao cliente. Os itens selecionados devem ser inseridos na planilha *Backlog* da *Sprint*, junto dos itens deve constar o número identificador da *Sprint*, a descrição da *Sprint*, que pode ser só o nome da *Sprint* (por exemplo, *Sprint 1*), a data estimada e a data realizada da entrega, e algum comentário sobre a *Sprint*.

### 5.3.2 Atividade “Especificar Caso de Uso”

A atividade “Especificar Caso de Uso” objetiva a especificação dos casos de uso selecionados para a *Sprint* atual. A Figura 7 apresenta o fluxo da atividade “Especificar Caso de Uso” e sua única tarefa.

Figura 7 – Fluxo da atividade “Especificar Caso de Uso”



Fonte: a própria autora.

#### 5.3.2.1 Tarefa “Especificar Caso de Uso”

A tarefa “Especificar Caso de Uso” objetiva especificar os casos de uso que serão desenvolvidos durante a *Sprint* atual. O Quadro 19 apresenta as informações referentes à tarefa “Especificar Caso de Uso”.

Quadro 19 – Representação da tarefa “Especificar Caso de Uso”

<b>Objetivo:</b> Esta tarefa consiste na especificação dos casos de uso baseado nos requisitos selecionados pela atividade Planejar <i>Sprint</i> .	
<b>Papéis principais:</b> Equipe de desenvolvimento	<b>Papéis secundários:</b> Líder técnico
<b>Entrada:</b> <i>Backlog</i> do Produto Caso de Uso CRUD Específico Caso de Uso CRUD Genérico Caso de Uso Relatório Específico Caso de Uso Relatório Genérico	<b>Saída:</b> Documento de Especificação de Caso de Uso
<b>Passos:</b> <ol style="list-style-type: none"> <li>1) Informações gerais sobre a especificação dos casos de uso</li> <li>2) Criar uma descrição breve do caso de uso</li> <li>3) Identificar atores</li> <li>4) Identificar fluxo básico do caso de uso</li> <li>5) Identificar fluxos alternativos do caso de uso</li> <li>6) Identificar pré-condições e pós-condições</li> <li>7) Identificar relacionamentos com outros casos de uso</li> <li>8) Construir diagrama do caso de uso</li> <li>9) Construção do documento de especificação de caso de uso</li> </ol>	

Fonte: a própria autora.

A seguir é apresentada a descrição dos passos desta tarefa.

**Passo 1 – Informações gerais sobre a especificação dos casos de uso:** A equipe de desenvolvimento irá especificar os casos de uso baseado nos requisitos selecionados para a *Sprint* e estes estão contidos no *Backlog* do Produto. Os casos de uso devem ser construídos baseados no *template* de casos de uso do tipo CRUD e Relatório, que podem ser Genérico ou Específico. O caso de uso CRUD Genérico tem como objetivo simplificar e apresentar um caso base de um CRUD simples e o caso de uso Relatório Genérico tem como objetivo simplificar e apresentar um caso base de um Relatório gerado na plataforma *Entities*. Para construir o caso de uso, deve estar atento se ele é do tipo CRUD, Relatório ou outro. Se ele for do tipo CRUD ou Relatório, deve-se utilizar os *templates* contidos neste processo, caso contrário deve ser construído um caso de uso novo. No documento de caso de uso CRUD ou Relatório específico, deve constar as informações específicas do caso de uso, como os dados a serem inseridos, qual o dado a ser utilizado na busca, entre outras. Os casos de uso do tipo CRUD Genérico e Relatório Genérico podem ser consultados nos Apêndices B e C respectivamente.

**Passo 2 – Criar uma descrição breve do caso de uso:** No início de cada caso de uso deve constar uma breve descrição explicando o seu objetivo.

**Passo 3 – Identificar atores:** Todos os atores envolvidos com o caso de uso devem ser identificados.

**Passo 4 – Identificar fluxo básico do caso de uso:** Deve ser identificado o cenário perfeito do caso de uso chamado de *'Happy Day'* ou fluxo básico. Neste cenário o caso de uso funciona normalmente sem nenhum erro.

**Passo 5 – Identificar fluxos alternativos do caso de uso:** Devem ser identificados fluxos que se desviam do fluxo básico, criando assim fluxos alternativos a ele.

**Passo 6 – Identificar pré-condições e pós-condições:** Devem ser identificadas pré-condições, que são condições que devem ser atendidas antes do início do caso de uso e pós-condições, que são condições que devem ser atendidas ao término do caso de uso.

**Passo 7 – Identificar relacionamentos com outros casos de uso:** Deve-se identificar se o caso de uso possui relacionamento com outros casos de uso.

**Passo 8 – Construir diagrama do caso de uso:** Deve-se construir um diagrama do caso de uso.

**Passo 9 – Construção do documento de especificação de caso de uso:** A partir das informações coletadas nos passos anteriores e dos modelos de caso de uso CRUD e Relatório genérico e específico, deve ser construído um documento de especificação para cada caso de uso.

### 5.3.3 Atividade “Implementação”

A atividade “Implementação” objetiva realizar atividades referentes à implementação dos casos de uso especificados na atividade anterior. A Figura 8 apresenta o fluxo da atividade “Implementação” e suas tarefas.

Figura 8 – Fluxo da atividade “Implementação”



Fonte: a própria autora.

#### 5.3.3.1 Tarefa “Preparar Ambiente”

A tarefa “Preparar Ambiente” objetiva preparar o ambiente em que o produto será desenvolvido. O Quadro 20 apresenta as informações referentes à tarefa “Preparar Ambiente”.

Quadro 20 – Representação da tarefa “Preparar Ambiente”

<b>Objetivo:</b> Esta tarefa consiste na preparação do ambiente de desenvolvimento.	
<b>Papéis principais:</b> Equipe de desenvolvimento Líder técnico	<b>Papéis secundários:</b> -
<b>Entrada:</b> Documento de Preparação do Ambiente <b>Entrada Opcional:</b> Tutorial de Instalação do <i>Entities</i>	<b>Saída:</b> Ambiente preparado
<b>Passos:</b> 1) Preparação do ambiente de desenvolvimento	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Preparação do ambiente de desenvolvimento:** A equipe de desenvolvimento deve preparar seu ambiente de trabalho, de acordo com a atividade a ser desenvolvida e linguagem de programação adotada no projeto, com a instalação das IDE’s necessárias para sua implementação. Essas informações estarão contidas no documento de preparação do ambiente, e este é específico de cada projeto.

### 5.3.3.2 Tarefa “Implementar Caso de Uso”

A tarefa “Implementar Caso de Uso” objetiva implementar todos os casos de uso selecionados na *Sprint* atual. O Quadro 21 apresenta as informações referentes à tarefa “Implementar Caso de Uso”.

Quadro 21 – Representação da tarefa “Implementar Caso de Uso”

<b>Objetivo:</b> Esta tarefa consiste na implementação dos casos de uso selecionados para a <i>Sprint</i> .	
<b>Papéis principais:</b> Equipe de desenvolvimento	<b>Papéis secundários:</b> Líder técnico
<b>Entrada:</b> Documento de Especificação de Caso de Uso	<b>Saída:</b> Caso de Uso Implementado
<b>Passos:</b> 1) Desenvolver caso de uso	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Desenvolver caso de uso:** A equipe de desenvolvimento deve implementar todos os casos de uso que foram selecionados para a *Sprint* atual e realizar teste funcional ao fim da implementação.

### 5.3.3.3 Tarefa “Gerar Release”

A tarefa “Gerar *Release*” objetiva que seja gerada uma *release* entregável do produto ao fim de cada *Sprint*. O Quadro 22 apresenta as informações referentes à tarefa “Gerar *Release*”.

Quadro 22 – Representação da tarefa “Gerar Release”

<b>Objetivo:</b> Esta tarefa consiste na geração de uma <i>release</i> entregável do produto.	
<b>Papéis principais:</b> Equipe de desenvolvimento	<b>Papéis secundários:</b> Líder técnico
<b>Entrada:</b> Caso de Uso Implementado	<b>Saída:</b> <i>Release</i> do Produto
<b>Passos:</b> 1) Gerar <i>release</i> do produto	

Fonte: a própria autora.

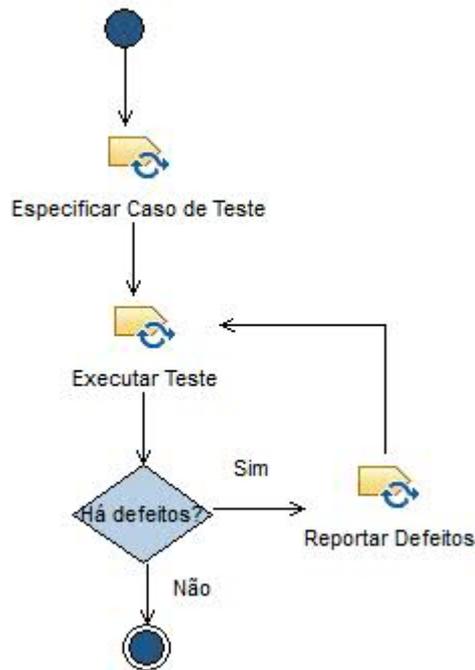
A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Gerar *release* do produto:** Todos os casos de uso implementados durante a *Sprint* devem ser unidos a fim de gerar uma *release* do produto potencialmente entregável. Devem ser realizados testes funcionais na *release* gerada.

### 5.3.4 Atividade “Testes”

A atividade “Testes” objetiva testar todos os casos de uso que foram construídos durante uma *Sprint* e a integração entre eles. A Figura 9 apresenta o fluxo da atividade “Testes” e suas tarefas.

Figura 9 – Fluxo da atividade “Testes”



Fonte: a própria autora.

#### 5.3.4.1 Tarefa “Especificar Caso de Teste”

A tarefa “Especificar Caso de Teste” objetiva a construção dos casos de teste. O Quadro 23 apresenta as informações referentes à tarefa “Especificar Caso de Teste”.

Quadro 23 – Representação da tarefa “Especificar Caso de Teste”

<b>Objetivo:</b> Essa tarefa consiste na criação dos casos de teste.	
<b>Papéis principais:</b> Testador	<b>Papéis secundários:</b> Equipe de desenvolvimento Líder técnico
<b>Entrada:</b> Backlog do Produto Documento de Especificação de Caso de Uso	<b>Saída:</b> Planilha de Teste
<b>Passos:</b> 1) Criar caso de teste	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Criar caso de teste:** Os casos de teste devem ser escritos baseados no documento de especificação de caso de uso e no *Backlog* do Produto, e devem ser inseridos na planilha de teste.

#### 5.3.4.2 Tarefa “Executar Teste”

A tarefa “Executar Teste” objetiva que os testes especificados na tarefa anterior sejam executados e seus resultados coletados. O Quadro 24 apresenta as informações referentes à tarefa “Executar Teste”.

Quadro 24 – Representação da tarefa “Executar Teste”

<b>Objetivo:</b> Esta tarefa consiste na execução dos testes no produto.	
<b>Papéis principais:</b> Testador	<b>Papéis secundários:</b> -
<b>Entrada:</b> Documento de Especificação de Caso de Uso Planilha de Teste Release do Produto	<b>Saída:</b> Planilha de Teste
<b>Passos:</b> 1) Executar os testes	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Executar os testes:** Os testes devem ser executados de acordo com os cenários descritos no documento de especificação de caso de uso. Todos os testes realizados devem ser inseridos na planilha de teste. Cada execução dos testes deve gerar uma nova aba na planilha.

#### 5.3.4.3 Tarefa “Reportar Defeitos”

A tarefa “Reportar Defeitos” objetiva que os defeitos encontrados sejam reportados para o líder técnico para que sejam corrigidos. O Quadro 25 apresenta as informações referentes à tarefa “Reportar Defeitos”.

Quadro 25 – Representação da tarefa “Reportar Defeitos”

<b>Objetivo:</b> Esta tarefa tem como objetivo reportar os defeitos encontrados durante a execução dos testes, para que os mesmos sejam corrigidos.	
<b>Papéis principais:</b> Líder técnico Testador	<b>Papéis secundários:</b> -
<b>Entrada:</b> Planilha de Teste	<b>Saída:</b> Planilha de Teste
<b>Passos:</b> 1) Reportar defeitos 2) Ações corretivas	

Fonte: a própria autora.

A seguir é apresentada a descrição dos passos desta tarefa.

**Passo 1 – Reportar defeitos:** Os defeitos ocorridos durante a execução dos testes devem ser inseridos na planilha de testes. A planilha de testes, que contém os problemas a serem corrigidos, deve ser adicionada ao *Redmine* em tarefa específica atribuída ao líder técnico do projeto. Ao serem corrigidos, os testes devem ser refeitos e o resultado deve ser inserido na planilha de teste.

**Passo 2 – Ações corretivas:** Ações corretivas devem ser tomadas em relação aos defeitos reportados.

### 5.3.5 Atividade “Revisão da Sprint”

A atividade “Revisão da *Sprint*” objetiva inspecionar o trabalho realizado durante a *Sprint* e adaptar o *Backlog* do Produto caso seja necessário. A Figura 10 apresenta o fluxo da atividade “Revisão da *Sprint*” e sua única tarefa.

Figura 10 – Fluxo da atividade “Revisão da *Sprint*”



Fonte: a própria autora.

### 5.3.5.1 Tarefa “Realizar Reunião de Revisão da *Sprint*”

A tarefa “Realizar Reunião de Revisão da *Sprint*” objetiva inspecionar a *release* do produto gerado durante a *Sprint* e adaptar o *Backlog* do Produto caso seja necessário. O Quadro 26 apresenta as informações referentes à tarefa “Realizar Reunião de Revisão da *Sprint*”.

Quadro 26 – Representação da tarefa “Realizar Reunião de Revisão da *Sprint*”

<b>Objetivo:</b> Esta tarefa tem como objetivo realizar a reunião de revisão da <i>Sprint</i> com o intuito de inspecionar o incremento e adaptar o <i>Backlog</i> do Produto se necessário.	
<b>Papéis principais:</b> Equipe de desenvolvimento Líder técnico	<b>Papéis secundários:</b> -
<b>Entrada:</b> <i>Backlog</i> do Produto <i>Release</i> do Produto	<b>Saída:</b> <i>Backlog</i> do Produto
<b>Passos:</b> 1) Realizar reunião de revisão da <i>Sprint</i>	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Realizar reunião de revisão da *Sprint*:** A Revisão da *Sprint* é executada no final da *Sprint* para inspecionar o incremento e adaptar o *Backlog* do Produto se necessário. Durante a reunião de Revisão da *Sprint*, a equipe de desenvolvimento e as partes interessadas colaboram sobre o que foi feito na *Sprint*. Com base nisso e em qualquer mudança no *Backlog* do Produto durante a *Sprint*, os participantes colaboram nas próximas coisas que precisam ser prontas. Esta é uma reunião informal, e a apresentação do incremento destina-se a motivar e obter comentários e promover a colaboração.

A Reunião de Revisão inclui os seguintes elementos:

- líder técnico identifica o que foi “Pronto” e o que não foi “Pronto”;
- a equipe de desenvolvimento discute o que foi bem durante a *Sprint*, quais problemas ocorreram dentro da *Sprint*, e como estes problemas foram resolvidos;
- a equipe de desenvolvimento demonstra o trabalho que está “Pronto” e responde as questões sobre o incremento;
- o líder técnico discute o *Backlog* do Produto tal como está. Ele (ou ela) projeta as prováveis datas de conclusão baseado no progresso até a data;

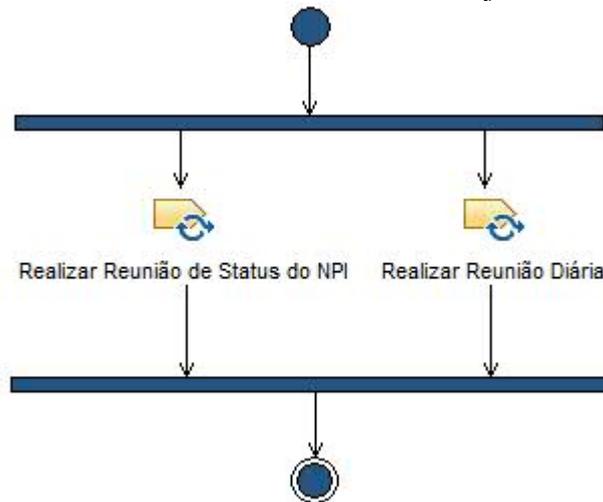
- o grupo todo colabora sobre o que fazer a seguir, e é assim que a Reunião de Revisão da *Sprint* fornece valiosas entradas para a Reunião de Planejamento da próxima *Sprint*.

O resultado da Reunião de Revisão da *Sprint* é um *Backlog* do Produto revisado que define o provável *Backlog* do Produto para a próxima *Sprint*. O *Backlog* do Produto pode também ser ajustado completamente para atender novas oportunidades.

#### 5.4 Atividade “Gerenciamento do Projeto”

A atividade “Gerenciamento do Projeto” é realizada após a atividade “Iniciar Projeto” e paralela as atividades “Requisitos”, “Gerenciamento de Configuração” e iteração “*Sprint*”. Ela é uma atividade de gerenciamento dos projetos. Dentro desta atividade contem tarefas que gerenciam a situação atual do projeto. Esse gerenciamento é algo bem simples. As tarefas são executadas de maneira independente e paralela uma a outra. A Figura 11 apresenta o fluxo da atividade “Gerenciamento do Projeto” e suas tarefas.

Figura 11 – Fluxo da atividade “Gerenciamento do Projeto”



Fonte: a própria autora.

##### 5.4.1 Tarefa “Realizar Reunião de Status do NPI”

A tarefa “Realizar Reunião de *Status* do NPI” objetiva reunir todos os integrantes dos projetos para discutir sobre os problemas que estão sendo enfrentados em cada projeto, com o intuito de se houverem problemas em comum, as equipes dos projetos possam se

auxiliar. O Quadro 27 apresenta as informações referentes à tarefa “Realizar Reunião de *Status* do NPI”.

Quadro 27 – Representação da tarefa “Realizar Reunião de *Status* do NPI”

<b>Objetivo:</b> A reunião de <i>status</i> do NPI é uma reunião semanal com todos os integrantes dos projetos, que tem como intuito compartilhar as dificuldades e os conhecimentos individuais entre os participantes.	
<b>Papéis principais:</b> Equipe de desenvolvimento Líder técnico Professor supervisor	<b>Papéis secundários:</b> -
<b>Entrada:</b> -	<b>Saída:</b> -
<b>Passos:</b> 1) Realizar reunião de <i>status</i> do NPI	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Realizar reunião de *status* do NPI:** No Núcleo de Práticas de Informática ocorrerá uma reunião por semana com todas as equipes de todos os projetos, para discutir os problemas que os integrantes estão tendo, e caso alguém já tenha passado por um problema semelhante, que ele possa auxiliar a equipe tirando as dúvidas. O ideal é que o problema seja resolvido na hora. Além disso, é uma oportunidade para os participantes trocarem conhecimentos que possam auxiliar no desenvolvimento dos projetos.

#### 5.4.2 Tarefa “Realizar Reunião Diária”

A tarefa “Realizar Reunião Diária” objetiva a realização de uma reunião diária curta com o intuito de discutir sobre as atividades a serem desenvolvidas e os seus impedimentos. O Quadro 28 apresenta as informações referentes à tarefa “Realizar Reunião Diária”.

Quadro 28 – Representação da tarefa “Realizar Reunião Diária”

<b>Objetivo:</b> Devem ser realizadas reuniões diárias com o intuito de que a equipe discuta sobre as atividades que estão sendo desenvolvidas e os seus impedimentos. A reunião deve durar no máximo 15 minutos.	
<b>Papéis principais:</b> Equipe de desenvolvimento Líder técnico	<b>Papéis secundários:</b> -
<b>Entrada:</b> -	<b>Saída:</b> <i>Backlog</i> do Produto
<b>Opcional:</b> <i>Backlog</i> do Produto	
<b>Passos:</b> 1) Realizar reunião diária	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Realizar reunião diária:** No início ou no fim do dia de trabalho das equipes, deve ocorrer uma reunião diária com o intuito de discutir com a equipe de desenvolvimento quais atividades foram realizadas desde a última reunião, quais serão realizadas e os possíveis impedimentos para a realização destas.

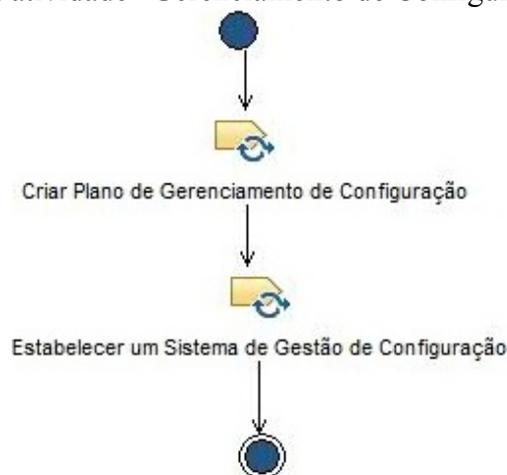
Esta reunião é feita para inspecionar o trabalho desde a última Reunião Diária, e prever o trabalho que deverá ser feito antes da próxima Reunião Diária. Ela deve ser conduzida pelo líder técnico e todos os participantes do projeto poderão participar. A reunião deve durar no máximo 15 minutos.

Caso sejam relatados impedimentos, estes devem ser registrados no *Backlog* do Produto como uma tarefa do tipo Impedimento e acompanhados pelo líder técnico e também podem ser repassados ao professor supervisor.

### 5.5 Atividade “Gerenciamento de Configuração”

A atividade “Gerenciamento de Configuração” é realizada após a atividade “Iniciar Projeto” e paralela às atividades “Requisitos”, “Gerenciamento do Projeto” e iteração “*Sprint*”. Ela é uma atividade de gerenciamento de configuração dos projetos. Esta atividade tem como objetivo gerenciar todos os artefatos do projeto e armazená-los de forma que todos os integrantes do projeto tenham acesso. A Figura 12 apresenta o fluxo da atividade “Gerenciamento de Configuração” e suas tarefas.

Figura 12 – Fluxo da atividade “Gerenciamento de Configuração”



Fonte: a própria autora.

### 5.5.1 Tarefa “Criar Plano de Gerenciamento de Configuração”

A tarefa “Criar Plano de Gerenciamento de Configuração” objetiva criar um plano de gerenciamento de configuração que deve conter todos os artefatos que serão gerenciados no projeto. O Quadro 29 apresenta as informações referentes à tarefa “Criar Plano de Gerenciamento de Configuração”.

Quadro 29 – Representação da tarefa “Criar Plano de Gerenciamento de Configuração”

<b>Objetivo:</b> O plano de gerenciamento de configuração descreve todas as atividades relacionadas ao gerenciamento de configuração a serem realizadas no decorrer do ciclo de vida do produto/projeto.	
<b>Papéis principais:</b> Líder técnico	<b>Papéis secundários:</b> Equipe de desenvolvimento
<b>Entrada:</b> -	<b>Saída:</b> Plano de Gerenciamento de Configuração
<b>Passos:</b> 1) Identificar itens de configuração 2) Atribuir identificadores únicos para os itens de configuração 3) Identificar o responsável por cada item de configuração 4) Criar plano de gerenciamento de configuração	

Fonte: a própria autora.

A seguir é apresentada a descrição dos passos desta tarefa.

**Passo 1 – Identificar itens de configuração:** A identificação de itens de configuração consiste da seleção, criação e especificação de:

- produtos que serão entregues ao cliente;
- produtos de trabalho internos selecionados;
- produtos adquiridos;
- ferramentas e outros bens de capital do ambiente de trabalho do projeto;
- outros itens que são utilizados na criação e descrição desses produtos de trabalho.

Exemplos de produtos de trabalho que podem fazer parte de um item de configuração:

- descrição de processos;
- requisitos;
- planos e procedimentos de teste;
- resultados de testes;
- diagramas;
- código-fonte.

**Passo 2 – Atribuir identificadores únicos para os itens de configuração:** Para cada item de configuração deve ser atribuído um identificador único. O identificador deve seguir o seguinte padrão: [PROJETO]-[TIPO]-EXTRA.EXTENSÃO. Exemplo: [NPI]-REQ-Especificacao.doc.

**Passo 3 – Identificar o responsável por cada item de configuração:** Cada item de configuração deve possuir um responsável.

**Passo 4 – Criar plano de gerenciamento de configuração:** O Plano de Gerenciamento de Configuração (GC) deve ser criado a partir das informações fornecidas nos passos anteriores. O Plano de GC deverá conter uma pequena descrição falando sobre tudo o que tem no plano; os responsáveis pelas atividades de GC; as ferramentas e ambientes que serão utilizados para desempenhar as funções de GC; e todos os itens de configuração identificados nos passos anteriores.

### 5.5.2 Tarefa “Estabelecer um Sistema de Gestão de Configuração”

A tarefa “Estabelecer um Sistema de Gestão de Configuração” objetiva o armazenamento de todos os itens de configuração em uma ferramenta com o propósito de gerenciar estes itens do projeto. O Quadro 30 apresenta as informações referentes à tarefa “Estabelecer um Sistema de Gestão de Configuração”.

Quadro 30 – Representação da tarefa “Estabelecer um Sistema de Gestão de Configuração”

<b>Objetivo:</b> Esta tarefa tem como objetivo selecionar uma ferramenta para gestão de configuração, armazenar nela os itens de configuração e gerenciá-los.	
<b>Papéis principais:</b> Líder técnico	<b>Papéis secundários:</b> Equipe de desenvolvimento
<b>Entrada:</b> Plano de Gerenciamento de Configuração	<b>Saída:</b> -
<b>Passos:</b> 1) Estabelecer um sistema de gestão de configuração	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Estabelecer um sistema de gestão de configuração:** O Sistema de Gestão de Configuração define as ferramentas para seu acesso, o ambiente de armazenamento e as diretrizes para criação e alteração de itens de configuração.

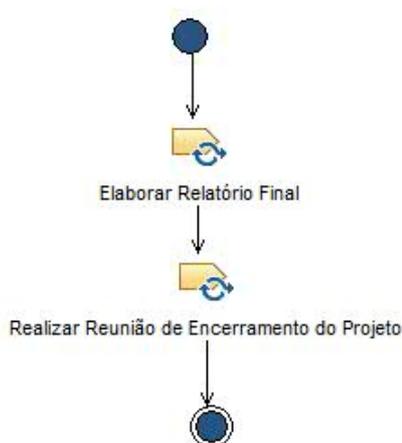
Cabe ao professor supervisor orientar os alunos sobre as ferramentas a serem utilizadas para controle de versão e armazenamento dos itens de configuração.

Ao ser decidido quais ferramentas serão utilizadas, deverá ser criado um ambiente de armazenamento, onde todos os itens de configuração identificados devem estar contidos e deve-se utilizar uma ferramenta para controle de versão.

## 5.6 Atividade “Encerrar Projeto”

A atividade “Encerrar Projeto” é a última atividade a ser executada no projeto. Ela é uma atividade de encerramento das tarefas e dos recursos alocados para os projetos. A Figura 13 apresenta o fluxo da atividade “Encerrar Projeto” e suas tarefas.

Figura 13 – Fluxo da atividade “Encerrar Projeto”



Fonte: a própria autora.

### 5.6.1 Tarefa “Elaborar Relatório Final”

A tarefa “Elaborar Relatório Final” objetiva que todos os integrantes do NPI elaborem um relatório final do estágio explanando sobre suas experiências. O Quadro 31 apresenta as informações referentes à tarefa “Elaborar Relatório Final”.

Quadro 31 – Representação da tarefa “Elaborar Relatório Final”

<b>Objetivo:</b> Esta tarefa tem como objetivo elaborar o relatório final do estágio.	
<b>Papéis principais:</b> Equipe de desenvolvimento	<b>Papéis secundários:</b> -
<b>Entrada:</b> -	<b>Saída:</b> Relatório Final
<b>Passos:</b> 1) Elaborar relatório final	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Elaborar relatório final:** Deve ser elaborado um relatório final o qual deve conter:

- dados de todos os integrantes da equipe;
- dados da instituição do estágio;
- descrever a empresa onde a equipe realizou o estágio;
- escrever um relato de experiência descrevendo como foi o estágio, as atividades mostradas no relato devem ser demonstradas em uma tabela com o cronograma destas e as lições aprendidas durante o estágio devem ser descritas, bem como as dificuldades encontradas e as oportunidades de melhoria identificadas;
- breve encerramento do relatório.

O *template* do relatório final deve servir de guia para a construção do relatório.

### 5.6.2 Tarefa “Realizar Reunião de Encerramento do Projeto”

A tarefa “Realizar Reunião de Encerramento do Projeto” objetiva reunir todos os integrantes do NPI com o intuito de encerrar as atividades e desalocar os recursos dos projetos, e obter o aceite do cliente quanto ao produto final. O Quadro 32 apresenta as informações referentes à tarefa “Realizar Reunião de Encerramento do Projeto”.

Quadro 32 – Representação da tarefa “Realizar Reunião de Encerramento do Projeto”

<b>Objetivo:</b> Esta tarefa tem como objetivo encerrar todas as atividades do projeto.	
<b>Papéis principais:</b> Cliente Equipe de desenvolvimento Líder técnico Professor supervisor	<b>Papéis secundários:</b> -
<b>Entrada:</b> Backlog do Produto Produto Final	<b>Saída:</b> -
<b>Passos:</b> 1) Realizar reunião de encerramento do projeto	

Fonte: a própria autora.

A seguir é apresentada a descrição do único passo desta tarefa.

**Passo 1 – Realizar reunião de encerramento do projeto:** O encerramento do projeto deve ser realizado visando obter um aceite do produto final do cliente, caso o produto

tenha sido concluído. Se necessário um termo de aceite deve ser formalizado. Todas as atividades do projeto devem ser finalizadas e os recursos devem ser desalocados.

### **5.7 Conclusão da Seção**

Esta Seção apresentou detalhadamente todas as atividades e tarefas do processo proposto. A próxima Seção apresenta os resultados obtidos, as dificuldades, lições aprendidas e oportunidades de melhoria identificados com a implantação do processo.

## 6 IMPLANTAÇÃO DO PROCESSO NO NPI

Esta Seção tem como objetivo apresentar os resultados obtidos com a implantação do processo em projetos pilotos no NPI. Esta Seção é composta de cinco subseções. A subseção 6.1 apresenta o *checklist* que foi aplicado nos projetos para medir a aderência da execução com o processo proposto. A subseção 6.2 apresenta a análise das entrevistas realizadas com os envolvidos na implantação do processo. A subseção 6.3 apresenta as dificuldades identificadas durante a execução do processo. A subseção 6.3 apresenta as lições aprendidas com a implantação do processo nos projetos pilotos. A subseção 6.5 apresenta as melhorias sugeridas pelos envolvidos. A subseção 6.6 apresenta a conclusão desta Seção.

### 6.1 Checklist de aderência ao processo proposto

O processo passou por diversas versões até chegar a uma versão final. Esta versão foi implantada em projetos pilotos do NPI. Durante a execução do processo, um *checklist* foi aplicado nos projetos para medir a aderência da execução com o processo proposto. As respostas obtidas no *checklist* são apresentadas nos Quadros 33, 34, 35 e 36. Cada quadro apresenta as perguntas e respostas de uma determinada atividade. A seguir, o Quadro 33 apresenta o *checklist* de aderência do processo do NPI na atividade Iniciar Projeto.

Quadro 33 – *Checklist* de aderência do processo do NPI – Atividade Iniciar Projeto

<b>Atividade: Iniciar Projeto</b>		
<b>Tarefa</b>	Prospectar Projetos	
<b>Perguntas</b>	O Documento de Visão foi elaborado pelo professor supervisor, de acordo com as informações obtidas nas reuniões com o cliente e possui todos os seus campos preenchidos corretamente, de forma clara e coerente, conforme <i>template</i> definido?	
<b>Saídas</b>	Documento de Visão	
<b>GAL</b>	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	O documento foi elaborado de acordo com reuniões com o cliente, mas essa atividade foi feita pelo líder técnico e equipe e apenas revisado pelo professor supervisor.
<b>GAP</b>	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	O documento de visão foi elaborado pela equipe e pelo líder técnico. O professor supervisor apenas revisou o documento.
<b>Tarefa</b>	Realizar Reunião de <i>Kickoff</i>	
<b>Perguntas</b>	A Reunião de <i>Kickoff</i> , para abertura do projeto, foi realizada com todos os participantes do NPI? Foram alocados os recursos humanos para cada projeto? O Documento de Visão de cada Projeto foi apresentado a equipe do Projeto? Foi criada uma ata de reunião, contendo todo o conteúdo da reunião e ele foi elaborado conforme <i>template</i> definido?	
<b>Saídas</b>	Ata de Reunião de <i>Kickoff</i>	
<b>GAL</b>	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	Sim. Sim. O Documento de Visão ainda não tinha sido construído, ele só foi feito depois de reuniões com os <i>stakeholders</i> . Foi criada uma ata de reunião, mas ela não foi elaborada conforme o <i>template</i> .
<b>GAP</b>	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	Sim. Sim. O documento de Visão somente foi construído depois de reuniões com os <i>stakeholders</i> . Foi criada uma ata de reunião, mas ela não foi elaborada conforme o <i>template</i> .
<b>Tarefa</b>	Elaborar Relatório Inicial	
<b>Perguntas</b>	O Relatório Inicial do Estágio foi elaborado por todos os integrantes do NPI e possui todos os seus campos preenchidos corretamente, conforme <i>template</i> definido?	
<b>Saídas</b>	Relatório Inicial	
<b>GAL</b>	<b>Resultado</b>	Não atende
	<b>Não Conformidades e/ou Observações</b>	Professor supervisor considerou o relatório inicial de estágio como não necessário como item a ser descrito no processo e não cobrou isso das equipes.
<b>GAP</b>	<b>Resultado</b>	Não atende
	<b>Não Conformidades e/ou Observações</b>	Professor supervisor considerou o relatório inicial de estágio como não necessário como item a ser descrito no processo e não cobrou isso das equipes.

Fonte: a própria autora.

A partir das respostas obtidas no *checklist*, podemos entender como ocorreu cada tarefa e algumas dificuldades enfrentadas.

Na atividade Iniciar Projeto, o professor supervisor prospectou os projetos e os apresentou aos alunos durante uma reunião de *Kickoff*. Nessa reunião foram alocados todos os recursos humanos de cada projeto. O professor supervisor definiu os líderes técnicos de cada projeto e alocou as equipes baseado no perfil profissional de cada um. Nessa reunião, uma ata

de reunião foi gerada, porém esta não foi criada baseado no *template* que consta no processo, pois o mesmo ainda não havia sido apresentado aos integrantes do NPI.

Durante esta atividade, algumas não conformidades ocorreram. Estas são explanadas a seguir.

O Documento de Visão, que no processo estava planejado para ser executado como saída na tarefa Prospectar Projetos e entrada na tarefa Realizar Reunião de *Kickoff*, foi realizado no início do projeto, porém o mesmo foi construído pelo líder técnico e pela equipe, baseado em reuniões com os *stakeholders*. Ele foi planejado para ser uma simples apresentação do projeto, porém o seu *template* exigia um nível alto de detalhes e ele acabou sendo utilizado como um documento onde todos os requisitos elicitados pela equipe foram inseridos. O *template* do Documento de Visão está disponível para consulta no Anexo A.

E a tarefa Elaborar Relatório Inicial não foi realizada, pois o professor supervisor não viu necessidade desta tarefa no processo do NPI. Ele afirmou durante a entrevista que achava que esta tarefa não deveria estar no processo, visto que futuramente o processo pode ser usado por pessoas que não estão fazendo estágio.

Seguindo o fluxo do processo, após a atividade Iniciar Projeto temos a atividade Requisitos, Gerenciamento do Projeto e Gerenciamento de Configuração. A seguir, o Quadro 34 apresenta o *checklist* de aderência do processo do NPI na atividade Requisitos. As outras atividades serão apresentadas na sequência.

Quadro 34 – Checklist de aderência do processo do NPI – Atividade Requisitos

Atividade: Requisitos		
<b>Tarefa</b>	Elicitar Requisitos	
<b>Perguntas</b>	<p>Todos os requisitos do projeto foram especificados pelo Líder Técnico juntamente com a equipe de desenvolvimento, tendo sido fornecidas informações como: documento de visão, necessidades identificadas através de entrevistas com o cliente, funcionalidades e restrições?</p> <p>O Documento de Especificação de Requisitos foi elaborado pelo líder técnico e a equipe e possui todos os seus campos preenchidos corretamente, de forma clara e coerente, conforme <i>template</i> definido?</p> <p>Obs1: A Especificação de Requisitos pode ir sendo construída ao longo das Sprints com relação apenas ao detalhamento das funcionalidades. Logo, todos os requisitos devem estar presentes na especificação, contendo pelo menos uma breve descrição.</p>	
<b>Saídas</b>	Documento de Especificação dos Requisitos	
GAL	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	<p>Os requisitos foram especificados através de reuniões com os <i>stakeholders</i>. O Documento de Visão foi construído nessa tarefa, sendo que todos os requisitos elicitados foram inseridos nele.</p> <p>O Documento de Especificação dos Requisitos não foi construído porque da forma como ele está no <i>template</i>, ele é somente uma lista de requisitos. O professor supervisor não viu necessidade de construí-lo sendo que os requisitos já estavam contidos no Documento de Visão.</p>
GAP	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	<p>Sim. O Documento de Especificação dos Requisitos não foi elaborado como proposto. Pois da forma como está estruturado o <i>template</i> do documento, ele é uma lista de requisitos. Não se viu necessidade de criar um documento a mais, já que essa lista de requisitos estará presente no <i>Backlog</i> do Produto.</p>
<b>Tarefa</b>	Validar Requisitos	
<b>Perguntas</b>	<p>Os requisitos especificados foram validadas pelo cliente?</p> <p>Houve alguma modificação nos requisitos ou adição e remoção de algum requisito?</p> <p>Se houve mudanças, estas foram registradas no Documento de Especificação dos Requisitos?</p> <p>Obs1: A validação pode ocorrer a partir de reunião ou email.</p>	
<b>Saídas</b>	Documento de Especificação dos Requisitos	
GAL	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	<p>A validação foi feita pelo <i>stakeholder</i>.</p> <p>As mudanças foram adicionadas no Documento de Visão.</p>
GAP	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	<p>A validação dos requisitos pelo cliente foi feita sim, mas através da apresentação do Documento de Visão, o qual continha todos os requisitos elicitados.</p>
<b>Tarefa</b>	Criar <i>Backlog</i> do Produto	
<b>Perguntas</b>	<p>O <i>Backlog</i> do Produto foi elaborado pelo líder técnico e a equipe, de acordo com o <i>template</i> definido e contém todos os requisitos especificados no Documento de Especificação de Requisitos?</p> <p>Obs1: O <i>Backlog</i> do Produto deve ser composto por histórias ou requisitos, ou seja, por tarefas que representam entregas efetivas, como funcionalidades, documentação, etc.</p>	
<b>Saídas</b>	<i>Backlog</i> do Produto	
GAL	<b>Resultado</b>	-
	<b>Não Conformidades e/ou Observações</b>	A equipe não chegou nessa tarefa até o fechamento deste trabalho.
GAP	<b>Resultado</b>	-
	<b>Não Conformidades e/ou Observações</b>	A equipe não chegou nessa tarefa até o fechamento deste trabalho.

Fonte: a própria autora.

Na atividade Requisitos, os requisitos dos projetos foram elicitados a partir de reuniões com *stakeholders*. As equipes dos dois projetos tiveram que se reunir mais de uma vez para poderem entender melhor os requisitos do sistema. Eles tiveram bastante dificuldade nessa etapa porque a maioria nunca tinha elicitado requisitos e nem tinha feito isso com *stakeholders* reais. O auxílio do professor supervisor foi de fundamental importância para a execução dessa atividade. Ele orientou e sanou as dúvidas em relação a elicitação dos requisitos e também de como preencher o *template* do Documento de Visão.

Na tarefa Validar Requisitos, a validação dos requisitos foi feita por meio de reunião, mostrando aos *stakeholders* o Documento de Visão com todos os requisitos elicitados e o *Roadmap* do projeto. Este último é uma espécie de mapa que visa organizar as metas de desenvolvimento do projeto. No *Roadmap* dos projetos, temos um mapeamento das funcionalidades e os seus respectivos requisitos, e estes estão organizados por entrega.

Nesta atividade, o processo foi atendido largamente, tendo somente uma não conformidade, que foi a não construção do documento de especificação dos requisitos. Porém, os requisitos foram levantados, especificados e validados. Portanto, a falta deste documento não comprometeu o bom andamento desta atividade e o seu objetivo foi alcançado. O *template* do Documento de Especificação dos Requisitos está disponível para consulta no Anexo B.

Paralelo à atividade Requisitos temos a atividade Gerenciamento do Projeto. A seguir, o Quadro 35 apresenta o *checklist* de aderência do processo do NPI nesta atividade.

Quadro 35 – *Checklist* de aderência do processo do NPI – Atividade Gerenciamento do Projeto

<b>Atividade: Gerenciamento do Projeto</b>		
<b>Tarefa</b>	Realizar Reunião de Status do NPI	
<b>Perguntas</b>	São realizadas reuniões semanais com todas as equipes de todos os projetos para discutir os problemas que os integrantes estão tendo?	
<b>Saídas</b>	-	
<b>GAL</b>	<b>Resultado</b>	Não atende
	<b>Não Conformidades e/ou Observações</b>	Não foram realizadas reuniões com todos os grupos do NPI. As reuniões ocorreram de modo individual entre o professor supervisor e cada equipe.
<b>GAP</b>	<b>Resultado</b>	Não atende
	<b>Não Conformidades e/ou Observações</b>	Não foram realizadas reuniões com todos os grupos do NPI. As reuniões ocorreram de modo individual entre o professor supervisor e cada equipe.
<b>Tarefa</b>	Realizar Reunião Diária	
<b>Perguntas</b>	São realizadas reuniões diárias no início ou fim do dia de trabalho das equipes, conduzidas pelo líder técnico com a participação dos membros da equipe? Foram relatados impedimentos? Os mesmos foram registrados no <i>Backlog</i> do Produto?	
<b>Saídas</b>	<i>Backlog</i> do Produto	
<b>GAL</b>	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	As reuniões diárias só ocorreram durante a construção do Documento de Visão, após isso, cada um dos integrantes começou a trabalhar isoladamente.
<b>GAP</b>	<b>Resultado</b>	Atende largamente
	<b>Não Conformidades e/ou Observações</b>	As reuniões diárias só ocorreram durante a construção do Documento de Visão, após isso, a equipe entrou em fase de estudo das ferramentas e não mais se reuniram.

Fonte: a própria autora.

Na atividade Gerenciamento do Projeto, as reuniões diárias aconteceram basicamente quando o projeto tinha algum artefato em construção, como o Documento de Visão. As reuniões estavam sendo realizadas com a presença do professor supervisor e pela sua iniciativa. As equipes em momento algum expressaram tomar a iniciativa de realizar reuniões diárias por conta própria. Acredita-se que isso aconteceu por falta de experiência dos líderes técnicos e o foco em desenvolver as atividades do projeto ocasionou que estes não realizaram muito o papel de líder.

Durante esta atividade, a tarefa Realizar Reunião de Status do NPI não atendeu ao processo, pois nenhuma reunião desse tipo foi realizada.

Paralelo às atividades Requisitos e Gerenciamento do Projeto temos a atividade Gerenciamento de Configuração. A seguir, o Quadro 36 apresenta o *checklist* de aderência do processo do NPI nesta atividade.

Quadro 36 – *Checklist* de aderência do processo do NPI – Atividade Gerenciamento de Configuração

<b>Atividade: Gerenciamento de Configuração</b>		
<b>Tarefa</b>	Criar Plano de Gerenciamento de Configuração	
<b>Perguntas</b>	Foram identificados itens de configuração? Foi atribuído um identificador único para cada item de configuração? Foi identificado um responsável para cada item de configuração? O Plano de Gerenciamento de Configuração foi utilizado?	
<b>Saídas</b>	Plano de Gerenciamento de Configuração	
<b>GAL</b>	<b>Resultado</b>	Atende totalmente
	<b>Não Conformidades e/ou Observações</b>	Os itens de configuração foram identificados e a sua nomenclatura foi padronizada, porém os responsáveis por cada documento não foram identificados. O Plano de Gerenciamento de Configuração foi utilizado.
<b>GAP</b>	<b>Resultado</b>	Atende totalmente
	<b>Não Conformidades e/ou Observações</b>	Os itens de configuração foram identificados e a sua nomenclatura foi padronizada, porém os responsáveis por cada documento não foram identificados. O Plano de Gerenciamento de Configuração foi utilizado.
<b>Tarefa</b>	Estabelecer um Sistema de Gestão de Configuração	
<b>Perguntas</b>	Foi definido uma ferramenta para realizar a gestão de configuração? Todos os itens de configuração identificados foram inseridos na ferramenta com o seu identificador único? Está sendo utilizada alguma ferramenta para controle de versão?	
<b>Saídas</b>	-	
<b>GAL</b>	<b>Resultado</b>	Atende totalmente
	<b>Não Conformidades e/ou Observações</b>	Foi selecionado repositório SVN para armazenar os itens de configuração e para realizar o controle de versão.
<b>GAP</b>	<b>Resultado</b>	Atende totalmente
	<b>Não Conformidades e/ou Observações</b>	Foi definido como ferramenta para realizar a gestão de configuração um repositório SVN. Os itens de configuração identificados foram inseridos no repositório. Sim, SVN.

Fonte: a própria autora.

Na atividade Gerenciamento do Projeto, os itens de configuração foram identificados, nomeados segundo o padrão estabelecido no Plano de Configuração, porém não foram identificados os responsáveis por cada item, porque os projetos somente geraram dois documentos. Um repositório foi criado e todos os itens identificados foram armazenados nele.

Esta atividade foi executada normalmente e atendeu totalmente ao processo. A não identificação dos responsáveis nesse ponto do projeto não chega a ser uma não conformidade, por isso que a atividade foi atendida totalmente.

Algo que ocorreu nos dois projetos e que não estava previsto no processo, foi uma auditoria para verificar diversos itens do sistema de gestão de configuração, como estrutura das pastas, nomenclatura dos arquivos, presença de arquivos numerados e a sua quantidade, presença de arquivos duplicados, quantidade de arquivos, código fonte e frequência de alterações no repositório. O resultado da auditoria foi enviado a todas as equipes com o intuito de que as não conformidades fossem corrigidas. Pretende-se incluir essa prática como melhoria em versões futuras do processo.

As atividades mostradas acima foram as únicas executadas dentro dos projetos, as outras foram definidas, porém por conta do tempo não foi possível executá-las. A seguir é apresentado uma análise das entrevistas realizadas com os envolvidos no processo do NPI.

## 6.2 Análise das entrevistas

Além do *checklist*, foi realizado uma entrevista com os líderes técnicos dos projetos e com o professor supervisor, com o intuito de entender como ocorreu a execução do processo e coletar a visão deles sobre o processo.

A partir das informações obtidas por meio do *checklist* e das entrevistas com o professor supervisor e os líderes técnicos podemos perceber que, o processo ainda está muito imaturo e que temos ainda muito mais a acrescentar para amadurecê-lo, porém em iniciativas de melhoria de processo isso é algo comum. Sempre terão partes do processo de desenvolvimento a serem melhoradas e aperfeiçoadas. Principalmente, porque esta é uma primeira iniciativa de melhoria.

Podemos perceber, inicialmente, que uma das principais deficiências desta iniciativa de melhoria foi que o processo não estava acessível para todos os integrantes do NPI, em um ambiente onde eles pudessem acessar e desfrutar das versões mais recentes do processo. Isso fez com que eles não se atentassem a seguir verdadeiramente o processo.

Além disso, o fato do processo ter sido definido baseado nas experiências do grupo de Sistemas de Informação (SI) do NPI e com o auxílio do seu professor supervisor, fez com que o processo tivesse um pouco da metodologia do professor e do grupo. O processo proposto não se adequou plenamente à metodologia do grupo de Engenharia de Software (ES) e do seu professor supervisor. Outro ponto observado é que o grupo de SI já tem maturidade e o grupo de ES está em sua primeira formação.

No início do projeto, o Documento de Visão foi pensado para ser uma simples apresentação do projeto, porém o seu *template* acabou por ficar complexo demais para esta etapa, o que acarretou no seu uso como um Documento de Requisitos, porém creio que o uso dele não implica no total descarte de um Documento de Especificação de Requisitos. Pois, poderá existir futuramente um projeto com características mais complexas e que precise de um documento mais completo, contendo requisitos funcionais, não funcionais e regras de negócio. Deve ser analisada a possibilidade de usar este último como um documento opcional, assim como foi sugerido pelo professor supervisor.

A atividade de Requisitos pode-se dizer que foi a mais seguida. As equipes fizeram reuniões com os *stakeholders* e a partir destas elicitaram os requisitos. Nesta etapa eles tiveram dificuldades, mas essas foram bem comuns de quem não tem experiência e de quem raramente executou essa tarefa com *stakeholders* reais, pois durante a vida acadêmica muitos não vivenciam esta etapa ou a executam muito pouco.

A validação dos requisitos ocorreu como esperado. O grupo apresentou os requisitos elicitados ao cliente e os validou, além disso, eles apresentaram uma espécie de cronograma do projeto, dividida por entregas, mas sem datas estimadas. A divisão das entregas foi algo interessante e poderá estar presente em versões futuras do processo.

Até o fechamento deste trabalho, um projeto não tinha chegado à tarefa Criar *Backlog* do Produto e o outro estava iniciando-a e esta sendo feita de forma automatizada na ferramenta *Redmine*. Essa forma de realizar o *Backlog* não foi pensada nesta iniciativa, mas é algo a ser visto em versões futuras. Veremos se é interessante automatizar este documento e descartá-lo ou se ele deve permanecer tal como está.

A atividade de Gerenciamento do Projeto acabou por ser deixada um pouco de lado durante a definição deste processo e não pode atingir o que se era esperado dela, no final esta atividade ficou simples, porém ela poderia ter sido melhor aproveitada se os líderes técnicos tivessem realizados as reuniões diárias com o intuito de gerenciar melhor as atividades que estavam sendo desenvolvidas no projeto. A não realização ou o pouco aproveitamento desta atividade ocorreu de certa forma pela falta de experiência dos líderes técnicos e do seu foco nas suas atividades individuais.

A atividade de Gerenciamento de Configuração foi bem aproveitada. Pela primeira vez o NPI terá um histórico de projetos, por conta da utilização de um repositório onde os projetos poderão ficar disponíveis para consultas futuras. Além disso, foi agregado conhecimento a cerca do uso de repositório. Com isso, a partir do próximo semestre esta atividade pode evoluir e pode-se adicionar tarefas que contemplem o gerenciamento de mudanças no projeto.

Algo interessante que foi utilizado no NPI e que não foi pensado neste processo inicial, foi à elaboração do perfil de cada integrante, o que auxiliou na melhor escolha das equipes dos projetos.

A seguir são apresentadas algumas dificuldades, lições aprendidas e oportunidades de melhoria identificadas durante a execução do processo.

### 6.3 Dificuldades

As dificuldades identificadas durante a execução do processo foram:

- as equipes tiveram dificuldades em construir o Documento de Visão. Esta foi ocasionada pela não intimidade deles com esse documento;
- a maioria dos integrantes dos projetos do NPI tinham pouco conhecimento em elicitação de requisitos. Isso fez com que esta etapa fosse muito demorada;
- como em diversas organizações, a maioria das pessoas não são proativas. Durante o desenvolvimento das atividades, eles desperdiçaram tempo e acabaram esperando somente pelo aval do professor supervisor;
- as reuniões diárias que ocorreram foram somente com a presença do professor supervisor. Eles não tiveram a iniciativa de fazê-la por si mesmos;
- os projetos onde o processo foi implantado eram novos e utilizaram tecnologias que até então as equipes não conheciam. O estudo de novas tecnologias acabou atrasando um pouco os projetos;
- os líderes técnicos desempenham além desse papel, o de integrante da equipe, com responsabilidades além de gerenciar o projeto. Pela pouca experiência, ocasionou que foi dada pouca prioridade as atividades de gerenciamento do projeto, como reuniões diárias para saber o status das atividades desenvolvidas por cada integrante do projeto;
- o uso de algumas ferramentas era algo novo para alguns integrantes dos projetos, porém essa tarefa não os atrapalhou;
- a universidade passou duas semanas sem internet, isso ocasionou um atraso nos projetos, pois este recurso é importante para gerenciar os projetos e também para utilização para estudo e consulta pelas equipes.

### 6.4 Lições aprendidas

Durante a execução do processo pudemos identificar algumas lições aprendidas, que foram:

- a inexperiência dos integrantes do projeto faz com que algumas tarefas precisem de mais tempo para serem realizadas;

- o pouco conhecimento em elicitação de requisitos fez com que os integrantes dos projetos demorassem muito na execução dessa atividade, atrasando o projeto. Um treinamento sobre os assuntos que eles pouco dominam, como este, teria ajudado bastante;
- a maioria das pessoas não é proativa. O ideal nos projetos é que tenha alguém para supervisionar os integrantes, fazendo com que as atividades fluam melhor e mais rápido;
- a adoção de novas tecnologias nos projetos somado a inexperiência dos integrantes fez com que os projetos demorassem muito. Parte disso se deu pela baixo nível de inglês deles, pois o material de estudo das tecnologias era nesta língua;
- a inexperiência dos líderes técnicos fez com que eles não realizassem o gerenciamento do projeto da melhor maneira;
- o pouco conhecimento dos integrantes do NPI em processo de software juntamente com questões culturais, fez com que eles não enxergassem a importância em se utilizar um processo de software;
- a mudança de professor supervisor do NPI pode influenciar o processo diretamente, pois cada um tem uma metodologia diferente;
- alunos tem dificuldades em se auto-gerenciar. Por terem diversas tarefas do projeto e da graduação, na maioria das vezes, o tempo não é bem aproveitado, ocasionando no não cumprimento ou má realização das atividades.

## 6.5 Oportunidades de melhoria

Durante a execução deste trabalho diversas melhorias foram sugeridas. A maioria delas foram incorporadas, e as outras foram dadas durante a implantação do processo em projetos pilotos no NPI. Durante a execução do processo foi sugerido:

- retirar as tarefas Elaborar Relatório Inicial e Elaborar Relatório Final, porque estas tarefas são específicas para quem está matriculado na disciplina de estágio e futuramente o processo pode vir a ser usado por pessoas que não estão fazendo o estágio;
- inserir um Termo de Abertura do Projeto, substituindo o Documento de Visão na atividade Iniciar Projeto;

- reformular o Documento de Visão para que ele fique adequado ao início do projeto. Deixando-o mais simples, contendo somente os dados obtidos durante a prospecção do projeto pelo professor supervisor. Ou inseri-lo em outra etapa do projeto, da maneira como está agora;
- anexar no processo exemplos de Documento de Visão já preenchidos para auxiliar o entendimento da equipe;
- avaliar a possível união dos Documento de Visão e do Documento de Especificação dos Requisitos e avaliar a possibilidade de ter o primeiro documento como obrigatório e o segundo como opcional;
- inserir como documento opcional na tarefa Validar Requisitos, uma ata de reunião para formalizar e registrar a validação dos requisitos com o cliente, caso essa seja realizada por meio de reunião;
- avaliar se vai ser utilizado o artefato *Backlog* do Produto ou se ele será automatizado inserindo suas informações na ferramenta *Redmine*, facilitando assim o seu controle. Além disso, adicionar uma aba ou tarefa de planejamento de entrega, bem semelhante ao *Roadmap*;
- gerenciar os projetos através da ferramenta *Redmine*;
- priorizar o planejamento dos testes em versões futuras do processo;
- inserir exemplos do Documento de Visão já preenchidos. Pois isso poderia ter ajudado os alunos a entenderem melhor como construí-lo;
- avaliar a possibilidade de união dos Documento de Visão e do Documento de Especificação dos Requisitos e a de ter o primeiro documento como obrigatório e o segundo como opcional;
- inserir como documento opcional uma ata de reunião para formalizar e registrar a validação dos requisitos com o cliente.

## 6.6 Conclusão da Seção

Esta Seção apresentou todos os resultados obtidos durante a implantação do processo em projetos pilotos do NPI, como o *checklist* que mediu a aderência ao processo proposto, um pouco da visão dos envolvidos sobre o processo, como ocorreu o andamento dos projetos, uma análise da aderência do processo e das entrevistas e por fim, algumas dificuldades, lições aprendidas e oportunidades de melhoria. A próxima Seção apresenta as conclusões acerca do trabalho realizado e os trabalhos futuros.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Nesta Seção são feitas as considerações finais acerca do trabalho realizado, focando nas contribuições, limitações e trabalhos futuros.

### 7.1 Considerações finais

Processos de software são fundamentais nas organizações que visam padronizar seu processo de desenvolvimento e obter maior qualidade tanto no seu processo, quanto no produto final. Por isso, muitas organizações estão em busca de definir seus processos. Porém essa não é uma tarefa simples. A definição e implantação de um processo de software precisa ser bem planejada e executada para que se consiga atingir os seus objetivos.

O trabalho proposto visou estabelecer um processo comum do Núcleo de Práticas de Informática da UFC – Campus Quixadá, incluindo boas práticas de modelos de qualidade de software reconhecidos no mercado.

O processo proposto foi construído a partir da captura de informações a respeito do desenvolvimento de software do NPI, coletado em reuniões com o professor supervisor e outros professores e através de uma entrevista com os líderes técnicos e o professor supervisor. Além disso, foram estudados alguns trabalhos de definição e implantação de processo de software no meio empresarial e acadêmico, com o intuito de conhecer as técnicas e os modelos, que eram utilizados na sua construção. Com as informações coletadas nas reuniões e nas entrevistas, podemos perceber algumas carências do NPI, baseado nisso foram selecionadas algumas boas práticas de modelos de qualidade de software a serem incorporadas no processo do NPI. Estas práticas foram selecionadas focando-se nas áreas de Requisitos, Gerência de Projetos e Gerência de Configuração, que foram as áreas identificadas como mais carentes.

O processo proposto foi construído baseado nas informações coletadas durante as reuniões e a entrevista, e nas boas práticas de modelos de qualidade de software selecionadas. A construção do processo passou por diversas versões até chegar a uma versão final. Esta foi executada em projetos pilotos dentro do NPI. Os projetos foram acompanhados e um *checklist* foi aplicado, com o intuito de checar a aderência da execução com o processo proposto. Ao final foi realizada uma entrevista com os líderes técnicos dos projetos, na qual o processo foi executado, e com o professor supervisor. A aderência do processo foi analisada, juntamente com as respostas obtidas durante as entrevistas.

A partir desta análise, podemos perceber algumas contribuições deste trabalho. Essas foram a análise da situação atual do desenvolvimento de software do NPI, a documentação inicial do processo de desenvolvimento de software do NPI e a inserção de algumas melhorias que antes não existiam, como gerenciamento efetivo da configuração dos projetos, histórico dos projetos no repositório, visão mais clara do projeto por meio do Documento de Visão e padronização de documentos.

O processo foi implantado em projetos pilotos do NPI, com o intuito de analisar a solução proposta, porém os projetos não avançaram a ponto de passar por todas ou boa parte das fases do processo proposto, com isso não foi possível identificar algumas outras contribuições que ele poderia ter trazido para o desenvolvimento de software do NPI.

Durante a execução do trabalho proposto nos deparamos com algumas limitações, entre elas podemos citar a demora no desenvolvimento dos projetos do NPI ocasionada pela falta de internet por duas semanas na universidade, estudo de novas tecnologias que foram utilizadas nos projetos e pouco conhecimento dos integrantes em elicitação de requisitos. Com isso, a execução do processo ficou um pouco prejudicada, pois não foi possível pela falta de tempo, passar por todas as fases, comprometendo assim o resultado final deste trabalho.

Apesar das adversidades, o processo trouxe algumas melhorias para as equipes do NPI. Estas são citadas a seguir:

- o processo define os passos para a execução das tarefas. Isso auxilia bastante no entendimento de como realizar uma tarefa;
- os documentos foram padronizados e possuem explicações simples de como ser preenchidos, o que facilita na sua construção;
- os envolvidos em cada tarefa estão bem definidos;
- os projetos estão mais organizados e mais acessíveis, pois os documentos importantes estão armazenados no repositório.

Este trabalho foi uma iniciativa inicial de processo e foi importante para que os envolvidos no NPI visualizem a necessidade de utilização de processo e também deem continuidade com outras iniciativas de melhorias de processo.

## **7.2 Trabalhos futuros**

Durante a execução do processo proposto foram identificadas algumas possibilidades de trabalhos futuros:

- analisar todas as propostas de melhoria sugeridas durante esta iniciativa e caso sejam interessantes, inserí-las em versões futuras do processo;
- tornar o processo acessível a todos os integrantes do NPI. Isso pode ser realizado através de uma página web ou a publicação em um repositório;
- planejar melhor tanto a definição como a implantação do processo;
- analisar melhor o desenvolvimento das atividades desenvolvidas nos dois grupos do NPI;
- entender melhor as necessidades e expectativas dos professores supervisores diante do processo;
- executar pequenos ciclos de melhoria focando em áreas específicas;
- realizar um melhor acompanhamento da execução do processo;
- realizar treinamentos de assuntos específicos que estão no processo e que os integrantes possuem pouco conhecimento;
- iniciar o gerenciamento do processo;
- automatizar o gerenciamento do processo.



## REFERÊNCIAS

- BARBOSA, E. F. *et al.* **Introdução ao Teste de Software**. In: Minicurso apresentado no XIV Simpósio Brasileiro de Engenharia de Software – SBES, 2000, João Pessoa – PB, p. 330 – 378. Disponível em: <<http://www.inf.ufpr.br/silvia/topicos/apostilaUSP.pdf.gz>>. Acesso em: 11 de junho de 2012.
- BORK, C. J. **Customização e Implantação de um Processo de Desenvolvimento de Software baseado no Rational Unified Process (RUP)**. 2003. Disponível em: <[http://www.bc.furb.br/docs/MO/2002/266445\\_1\\_1.pdf](http://www.bc.furb.br/docs/MO/2002/266445_1_1.pdf)>. Acesso em: 16 de junho de 2013.
- CARVALHO, A. E. S.; TAVARES, H. C.; CASTRO, J. B. **Uma Estratégia para Implantação de uma Gerencia de Requisitos Visando a Melhoria Dos Processos de Software**. In: IV Workshop em Requisitos, 2001, Buenos Aires, Argentina. Disponível em: <[http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER01/carvalho.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER01/carvalho.pdf)>. Acesso em: 16 de junho de 2013.
- CATUNDA, E. *et al.* **Implementação do Nível F do MR-MPS com Práticas Ágeis do Scrum em uma Fábrica de Software**. In: X Simpósio Brasileiro de Qualidade de Software, 2011, Curitiba. Disponível em: <[http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2011/SBQS2011-RE10\\_82940.pdf](http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2011/SBQS2011-RE10_82940.pdf)>. Acesso em: 21 de junho de 2013.
- CINTRA, C. C. **A Implantação de um Processo de Engenharia de Requisitos Baseado no Processo Unificado da Rational (RUP) Alcançando Nível 3 de Maturidade da Integração de Modelos de Capacidade e Maturidade (CMMI) Incluindo a Utilização de Práticas de Métodos Ágeis**. 2006. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/8128/000568343.pdf?sequence=1>>. Acesso em: 10 de julho de 2013.
- CORGOSINHO, C. C. Como Iniciar e Acompanhar um Programa de Implantação do MPS.BR. **ProQualiti**, Recife, v. 2, p. 23 – 28, nov. 2006. Disponível em: <[http://www.tconibo.org/adega/revista\\_nov\\_2006pdf.pdf](http://www.tconibo.org/adega/revista_nov_2006pdf.pdf)>. Acesso em: 10 de julho de 2013.
- D'ÁVILA, M. **PMBOK e Gerenciamento de Projetos**. Revisão 5. 2011. Disponível em: <<http://www.mhavila.com.br/topicos/gestao/pmbok.html>>. Acesso em: 5 de novembro de 2012.
- DINSMORE, P. C.; CAVALIERI, A. **Como Se Tornar Um Profissional Em Gerenciamento de Projetos**. 4 ed. Rio de Janeiro: Qualitymark Editora, 2011.
- ECLIPSE FOUNDATION. **Eclipse Process Framework Composer**. Versão 1.5.1.3. 2011. Disponível em: <<http://www.eclipse.org/epf/>>. Acesso em: 08 de junho de 2012.
- FALBO, R. A.; BARCELLOS, M. P. **Engenharia de Software**. 2011. Disponível em: <<http://www.inf.ufes.br/~monalessa/PaginaMonalessa-NEMO/ES/NotasDeAula-EngSoftware-EngComp-Parte-I.pdf>>. Acesso em: 28 de junho de 2013.

FERREIRA, A. I. F. *et al.* **"Applying ISO 9001:2000, MPS.BR and CMMI to Achieve Software Process Maturity: BL Informatica's Pathway"**. In: 29th Int. Conference on Software Engineering (ICSE), 2007, Minneapolis, USA, p. 642 – 651.

GABRIEL, E. R. **A implantação do Processo de Garantia da Qualidade de Produto de Software baseado no MPS.BR**. 2009. Monografia (Curso de Tecnologia em Informática) – Centro Tecnológico da Zona Leste, Faculdade de Tecnologia da Zona Leste, São Paulo, 2009. Disponível em: <<http://fateczl.edu.br/TCC/2009-1/tcc-17.pdf>>. Acesso em: 16 de junho de 2013.

HICKS, M.; FOSTER, J. S. **Adapting Scrum to Managing a Research Group**. 2010. Disponível em: <<http://www.cs.umd.edu/~mwh/papers/score.pdf>>. Acesso em: 12 de junho de 2012.

KALINOWSKI, M. *et al.* **MPS.BR: Promovendo a Adoção de Boas Práticas de Engenharia de Software pela Indústria Brasileira**. In: XIII Congresso Iberoamericano em Engenharia de Software. 2010. Cuenca, Equador. Disponível em: <[http://www.softex.br/portal/softexweb/uploadDocuments/CIBSE2010\\_MPSBR\\_CameraReady.pdf](http://www.softex.br/portal/softexweb/uploadDocuments/CIBSE2010_MPSBR_CameraReady.pdf)>. Acesso em: 12 de junho de 2012.

KOSCIANSKI, A.; SOARES, M. S. **Qualidade de Software**. 2.ed. São Paulo: Editora Novatec, 2007.

KRUCHTEN, P. **Introdução ao RUP – Rational Unified Process**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2003.

MCFEELEY, B. **IDEAL: A User's Guide for Software Process Improvement**. 1996. CMU/SEI-96-HB-001. Pittsburgh, Software Engineering Institute.

MENDES, F. F. *et al.* **Implantação de Melhoria de Processos em um Setor de Produção de Software de uma Universidade Federal**. In: IX Simpósio Brasileiro de Qualidade de Software, 2010, Belém – PA. Disponível em: <[http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2010/RL11\\_fabiana\\_mendes.pdf](http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2010/RL11_fabiana_mendes.pdf)>. Acesso em: 11 de junho de 2012.

MENDES, F. F.; ALMEIDA, J. N.; JUNIOR, E. A. **Experiência de Implantação de Melhoria de Processos de Software em um Laboratório de Pesquisa**. In: VII Workshop Anual do MPS, 2011, Campinas – SP, Anais, p. 114 – 123. Disponível em: <[http://www.softex.br/portal/softexweb/uploadDocuments/Anais%20WAMPS%202011\(2\).pdf](http://www.softex.br/portal/softexweb/uploadDocuments/Anais%20WAMPS%202011(2).pdf)>. Acesso em: 16 de junho de 2013.

NUNES, E. D. *et al.* **MPS.BR Nível E - Uma Avaliação em Verde e Amarelo**. In: V Simpósio Brasileiro de Qualidade de Software – SBQS, 2006, Vila Velha – ES, Anais, p. 318 – 325.

PISKE, O. R. **RUP – Rational Unified Process**. 2003. Disponível em: <[http://www.angusyong.org/arquivos/artigos/trabalho\\_rup.pdf](http://www.angusyong.org/arquivos/artigos/trabalho_rup.pdf)>. Acesso em: 16 de junho de 2013

PMI . **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBoK)**. 4. ed. Pennsylvania: PMI, 2008.

SALGADO, A. et al. **Aplicação de um Processo Ágil para Implantação de Processos de Software baseado em Scrum na Chemtech**. In: IX Simpósio Brasileiro de Qualidade de Software, 2010, Belém – PA. Disponível em: <<http://nemo.inf.ufes.br/files/SCRUMeCMMI3naChemtech-2010.pdf>>. Acesso em: 21 de junho de 2013.

SANTOS, G. et al. **SPI-KM - Lessons Learned from Applying a Software Process Improvement Strategy Supported by Knowledge Management**, Product-Focused Software Process Improvement. 2007.

SANTOS, G. et al. **Indicadores da Implementação do Nível E do MR-MPS em uma Instituição de Pesquisa**. In: VIII Simpósio Brasileiro de Qualidade de Software, 2009, Ouro Preto – MG, Anais, p. 382 – 389. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2009/033.pdf>>. Acesso em: 10 de julho de 2013.

SCHEID, M. et al. **Implantação do MR-MPS Nível E no Centro de Computação da Aeronáutica de São José dos Campos**. 2007 Disponível em: <[http://www.softex.br/portal/softexweb/uploaddocuments/\\_mpsbr/t6-cca-we.pdf](http://www.softex.br/portal/softexweb/uploaddocuments/_mpsbr/t6-cca-we.pdf)>. Acesso em: 12 de junho de 2012.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum**. Outubro, 2011. Disponível em: <<http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20Portuguese%20BR.pdf#zoom=100>>. Acesso em: 1 de julho de 2013.

SEI. **CMMI for Development**. v. 1.3, 2010. Disponível em: <<http://www.sei.cmu.edu/reports/10tr033.pdf>>. Acesso em: 12 de junho de 2012.

SILVA, T. et al. **Implantação do Nível F do MR-MPS Combinando Características do Processo Unificado com Práticas SCRUM**. In: VII Workshop Anual do MPS, 2011, Campinas – SP, Anais, p. 54 – 61. Disponível em: <[http://www.softex.br/portal/softexweb/uploadDocuments/Anais%20WAMPS%202011\(2\).pdf](http://www.softex.br/portal/softexweb/uploadDocuments/Anais%20WAMPS%202011(2).pdf)>. Acesso em: 16 de junho de 2013.

SOFTEX. **Melhoria de Processo de Software Brasileiro: Guia Geral**. 2011. Disponível em: <[http://www.softex.br/mpsbr/\\_guias/guias/MPS.BR\\_Guia\\_Geral\\_2011.pdf](http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2011.pdf)>. Acesso em: 12 de junho de 2012.

SOMMERVILLE, I. **Engenharia de Software**. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

TSUKUMO, A. N. et al. Lições aprendidas na aplicação do Método Coop-MPS para Projetos Cooperativos de Melhoria de Processo de Software com MPS.BR. **ProQualiti**, Recife, v. 2, p. 51 – 56, nov. 2006. Disponível em: <[http://www.tconibo.org/adega/revista\\_nov\\_2006pdf.pdf](http://www.tconibo.org/adega/revista_nov_2006pdf.pdf)>. Acesso em: 10 de julho de 2013.

WTHREEX. RUP 2002.05.00 Portugues. v. 5.0. 2002. Disponível em: <<http://www.wthreex.com/rup/portugues/index.htm>>. Acesso em: 10 de julho de 2013.



# APÊNDICES



**APÊNDICE A**  
**CHECKLIST DE ADERÊNCIA DO**  
**PROCESSO DO NPI**



Sub processo	ID	Atividades do Processo	Saídas	Resultado	Não Conformidades e/ou Observações	Considerações do Scrum Master	Considerações do QA
Iniciar Projeto	1	O Documento de Visão foi elaborado pelo professor supervisor, de acordo com as informações obtidas nas reuniões com o cliente e possui todos os seus campos preenchidos corretamente, de forma clara e coerente, conforme template definido?	Documento de Visão				
	2	A Reunião de Kickoff, para abertura do projeto, foi realizada com todos os participantes do NPI? Foram alocados os recursos humanos para cada projeto? O Documento de Visão de cada Projeto foi apresentado a equipe do Projeto? Foi criada uma ata de reunião, contendo todo o conteúdo da reunião e ele foi elaborado conforme template definido?	Ata de Reunião de Kickoff				
	3	O Relatório Inicial do Estágio foi elaborado por todos os integrantes do NPI e possui todos os seus campos preenchidos corretamente, conforme template definido?	Relatório Inicial				

Subprocesso	ID	Atividades do Processo	Saídas	Resultado	Não Conformidades e/ou Observações	Considerações do Scrum Master	Considerações do QA
Requisitos	4	Todos os requisitos do projeto foram especificados pelo Líder Técnico juntamente com a equipe de desenvolvimento, tendo sido fornecidas informações como: documento de visão, necessidades identificadas através de entrevistas com o cliente, funcionalidades e restrições?O Documento de Especificação de Requisitos foi elaborado pelo líder técnico e a equipe e possui todos os seus campos preenchidos corretamente, de forma clara e coerente, conforme template definido?Obs1: A Especificação de Requisitos pode ir sendo construída ao longo das sprints com relação apenas ao detalhamento das funcionalidades. Logo, todos os requisitos devem estar presentes na especificação, contendo pelo menos uma breve descrição.	Documento de Especificação dos Requisitos				
	5	Os requisitos especificados foram validadas pelo cliente? Houve alguma modificação nos requisitos ou adição e remoção de algum requisito? Se houve mudanças, estas foram registradas no Documento de Especificação dos Requisitos? Obs1: A validação pode ocorrer a partir de reunião ou email.	Documento de Especificação dos Requisitos				
	6	O Backlog do Produto foi elaborado pelo líder técnico e a equipe, de acordo com o template definido e contem todos os requisitos especificados no Documento de Especificação de Requisitos? Obs1: O Product Backlog deve ser composto por histórias ou requisitos, ou seja, por tarefas que representam entregas efetivas, como funcionalidades, documentação, etc.	Backlog do Produto				

Sub processo	ID	Atividades do Processo	Saídas	Resultado	Não Conformidades e/ou Observações	Considerações do Scrum Master	Considerações do QA
Sprint	Planejar Sprint	7	Foi feita reunião da sprint com a presença do líder técnico e da equipe de técnica para selecionar os casos de uso que serão desenvolvidos na sprint atual?	Backlog do Produto			
		8	Os itens selecionados foram inseridos na planilha Backlog da Sprint, com respectivos numero identificador e descrição da Sprint, e data estimada da entrega?	Backlog do Produto			
	Especificar Caso de Uso	9	Em cada caso de Uso consta uma breve descrição do seu objetivo?	Documento de Especificação de Caso de Uso			
		10	Foram identificados os atores dos casos de uso?				
		11	Foram identificados os fluxos básicos e alternativos dos casos de uso?				
		12	Foram identificadas pré-condições e pós condições?				
		13	Foram identificados relacionamentos com outros casos de uso?				
		14	Foi construído diagrama de caso de uso?				
		15	Foi construído documento de especificação de caso de uso?				
	16	Os casos de uso foram construídos baseados nos templates do tipo CRUD e Relatório Específico e Genérico?					
	Implementação	17	A equipe de desenvolvimento preparou seu ambiente de trabalho de acordo com a atividade a ser desenvolvida e linguagem de programação adotada? Foi desenvolvido um documento de preparação do ambiente para auxiliar os integrantes da equipe na preparação do seu ambiente de trabalho?	Ambiente Preparado			
		18	Foram implementados todos os casos de uso que foram selecionados para a sprint atual, com teste funcional ao fim da implementação?	Caso de Uso Implementado			
		19	Os casos de uso implementados durante a Sprint foram unidos a fim de gerar uma release do produto potencialmente entregável, tendo realizado testes funcionais na release gerada?	Release do Produto			

Sub processo	ID	Atividades do Processo	Saídas	Resultado	Não Conformidades e/ou Observações	Considerações do Scrum Master	Considerações do QA	
Sprint	Testes	20	Foram escritos casos de testes baseados no Documento de especificação dos casos de uso e no <i>Backlog</i> do Produto?	Planilha de Teste				
		21	Os testes foram executados de acordo com os cenários descritos no documento de especificação de caso de uso?					
		22	Os testes realizados foram inseridos na Planilha de Teste , gerando uma nova aba na planilha?					
		23	Defeitos ocorridos durante a execução dos testes foram inseridos na Planilha de Teste?					
		24	A Planilha de Teste foi adicionada ao <i>Redmine</i> em tarefa específica atribuída ao líder técnico do projeto?					
		25	Os testes foram refeitos e os resultados inseridos na Planilha de Teste em uma nova aba?					
	26	Ações corretivas foram tomadas em relação aos defeitos reportados?						
	Revisão da Sprint	27	Foi realizada Reunião de Revisão da Sprint?	<i>Backlog</i> do Produto				
		28	Na reunião o Líder Técnico identificou o que está "Pronto" e o que não está?					
		29	Foi discutido pela Equipe de Desenvolvimento o que foi bem e os problemas ocorridos durante a Sprint ?					
		30	A equipe de desenvolvimento demonstrou o trabalho em estado "Pronto", respondendo questões sobre o incremento?					
		31	Foram projetadas prováveis datas de conclusão baseado no progresso até a data?					
32		O <i>Backlog</i> do Produto foi revisado?						

Sub processo	ID	Atividades do Processo	Saídas	Resultado	Não Conformidades e/ou Observações	Considerações do Scrum Master	Considerações do QA
Gerenciamento do Projeto	33	São realizadas reuniões semanais com todas as equipes de todos os projetos para discutir os problemas que os integrantes estão tendo?	N/A				
	34	São realizadas reuniões diárias no início ou no fim do dia de trabalho das equipes, conduzidas pelo líder técnico com a participação de todos os membros da equipe?	Backlog do Produto				
	35	Foram relatados impedimentos? Os mesmos foram registrados no <i>Backlog</i> do Produto?					
Gerenciamento de Configuração	36	Foram identificados itens de configuração? Essa identificação consiste em selecionar, criar e especificar os seguintes itens: - Produtos que serão entregues ao cliente. - Produtos de trabalho internos selecionados. - Produtos adquiridos. - Ferramentas e outros bens de capital do ambiente de trabalho do projeto. - Outros itens que são utilizados na criação e descrição desses produtos de trabalho.	Plano de Gerenciamento de Configuração				
	37	Foi atribuído um identificador único para cada item de configuração? O identificador deve seguir o seguinte padrão: [PROJETO]-[TIPO]-EXTRA.EXTENSÃO Exemplo: [NPI]-REQ-Especificacao.doc					
	38	Foi identificado um responsável para cada item de configuração?					
	39	O Plano de Gerenciamento de Configuração foi utilizado?					
	40	Foi definido uma ferramenta para realizar a gestão de configuração? Todos os itens de configuração identificados foram inseridos na ferramenta com o seu identificador único? Está sendo utilizada alguma ferramenta para controle de versão?	N/A				

Sub processo	ID	Atividades do Processo	Saídas	Resultado	Não Conformidades e/ou Observações	Considerações do Scrum Master	Considerações do QA
Encerrar Projeto	42	Foi elaborado relatório final contendo: - Dados de todos os integrantes da equipe; - Dados da instituição do estágio; - Descrever a empresa onde a equipe realizou o estágio; - Escrever um relato de experiência descrevendo como foi o estágio, as atividades mostradas no relato devem ser demonstradas em uma tabela com o cronograma destas e as lições aprendidas durante o estágio devem ser descritas, bem como as dificuldades encontradas e as oportunidades de melhoria identificadas. - Breve encerramento do relatório.	Relatório Final				
	43	Foi realizada Reunião de Encerramento do Projeto? Caso o projeto tenha sido finalizado, o produto final teve aceite do cliente? Foi elaborado um termo de aceite ou algum outro documento que formalizasse o aceite do cliente? Os recursos do projeto foram desalocados? As atividades do projeto foram finalizadas?	Backlog do Produto/ Produto Final				

# **APÊNDICE B**

## **CASO DE USO CRUD GENÉRICO**





UNIVERSIDADE  
FEDERAL DO CEARÁ

## ESPECIFICAÇÃO DE CASOS DE USO

**Caso de uso Manter Entidade**

**Projeto X – Sistema X**

**RESERVADO**

<b>Responsável:</b> Enyo José T. Gonçalves	
<b>Elaborador(es):</b>	<b>e-mail</b>
Micaelly P. Soares	micaellyps@gmail.com

## HISTÓRICO

<b>Data</b>	<b>Versão</b>	<b>Responsável</b>	<b>Alteração</b>
20/10/2012	01	Micaelly P. Soares	- Criação do documento

## ÍNDICE

1	INTRODUÇÃO .....	4
1.1	Objetivos .....	4
1.2	Público Alvo deste Documento .....	4
2	CASO DE USO MANTER ENTIDADE .....	4
2.1	Descrição .....	4
2.2	Atores Envolvidos.....	4
2.3	Fluxo Básico ( <i>Happy Day</i> ).....	4
2.3.1	Criar Entidade.....	4
2.3.2	Pesquisar Entidade .....	5
2.3.3	Atualizar Entidade .....	5
2.3.4	Excluir Entidade.....	5
2.4	Fluxos Alternativos ( <i>Exceptions</i> ).....	5
2.4.1	Cancelar Inclusão da Entidade.....	5
2.4.2	Cancelar Atualização da Entidade .....	5
2.4.3	Cancelar Exclusão da Entidade .....	5
2.4.4	Dados Obrigatórios Não Informados .....	5
2.4.5	Entidade Duplicada .....	6
2.5	Requisitos Especiais .....	6
2.6	Pré-Condições.....	6
2.7	Pós-Condições.....	6
2.8	Relacionamento com Outros Casos de Uso .....	6
2.8.1	Pontos de Inclusão .....	6
2.8.2	Pontos de Extensão .....	6

# ESPECIFICAÇÃO DE CASOS DE USO

## 1 INTRODUÇÃO

### 1.1 Objetivos

Este documento tem como objetivo detalhar o caso de uso manter Entidade.

### 1.2 Público Alvo deste Documento

- Equipe de Desenvolvimento
- Caso tenha mais algum público alvo específico, estes devem ser detalhados no caso de uso específico.

## 2 CASO DE USO MANTER ENTIDADE

### 2.1 Descrição

Este caso de uso se responsabiliza pela criação, pesquisa, atualização e exclusão da entidade.

### 2.2 Atores Envolvidos

- Os atores devem estar detalhados no caso de uso específico.

### 2.3 Fluxo Básico (*Happy Day*)

1. Este caso de uso inicia quando o usuário seleciona a opção manter Entidade;
2. O sistema requisita que o usuário especifique a operação que deseja realizar.
3. Uma vez que o usuário solicite executar uma das funções desejadas (criar, pesquisar, atualizar e excluir), um dos seguintes sub-fluxos é executado:
  - Se o usuário desejar “Criar”, o sub-fluxo 2.3.1 Criar Entidade é executado;
  - Se o usuário desejar “Pesquisar”, o sub-fluxo 2.3.2 Pesquisar Entidade é executado;
  - Se o usuário solicitar “Atualizar”, o sub-fluxo 2.3.3 Atualizar Entidade é executado.
  - Se o usuário solicitar “Excluir”, o sub-fluxo 2.3.4 Excluir Entidade é executado.

#### 2.3.1 Criar Entidade

1. Este sub-fluxo inicia quando o usuário seleciona a opção Criar Entidade.
2. O sistema exibe os campos da Entidade e permite ao usuário preenchê-los. A descrição e o detalhamento dos campos deve estar contido no caso de uso específico.
3. O usuário preenche os campos e solicita salvar o cadastro da Entidade.
4. O sistema critica os campos.
5. O sistema exibe uma mensagem de confirmação.

### 2.3.2 Pesquisar Entidade

1. Este fluxo inicia quando o usuário preenche o filtro e solicita a busca;
2. O sistema exibe as informações da Entidade de acordo com a busca requisitada;
3. O usuário seleciona um dos itens do resultado da busca;
4. Uma vez que o usuário solicite executar uma das funções desejadas da Entidade (atualizar ou excluir), um dos seguintes sub-fluxos é executado:
  - Se o usuário solicitar “Atualizar” o sub-fluxo 2.3.3 Atualizar Entidade é executado;
  - Se o usuário solicitar “Excluir”, o sub-fluxo 2.3.4 Excluir Entidade é executado.

### 2.3.3 Atualizar Entidade

1. Este fluxo inicia quando o usuário solicita a atualização dos dados da Entidade selecionada.
2. O sistema exibe os campos da Entidade já preenchidos com os dados e permite ao usuário alterá-los.
3. O usuário altera os campos e solicita salvar o cadastro da Entidade.
4. O sistema critica os campos.
5. O sistema exibe uma mensagem de confirmação.
6. O caso de uso retorna para o passo 2 do fluxo básico.

### 2.3.4 Excluir Entidade

1. Este sub-fluxo inicia quando o usuário solicita a exclusão da Entidade selecionada.
2. O sistema solicita confirmação da exclusão.
3. O usuário confirma a exclusão.
4. O sistema exibe uma mensagem de confirmação.

## 2.4 Fluxos Alternativos (*Exceptions*)

### 2.4.1 Cancelar Inclusão da Entidade

1. Se no passo 3 do sub-fluxo 2.3.1 Criar Entidade, o usuário solicitar cancelar a inserção da Entidade, o caso de uso retorna para o passo 2 do sub-fluxo 2.3.1 Criar Entidade.

### 2.4.2 Cancelar Atualização da Entidade

1. Se no passo 3 do sub-fluxo 2.3.3 Atualizar Entidade, o usuário solicitar cancelar a atualização da Entidade, o sistema desconsidera as atualizações da Entidade atual;

### 2.4.3 Cancelar Exclusão da Entidade

1. Se no passo 2 do sub-fluxo 2.3.4 Excluir Entidade o usuário solicitar cancelar a exclusão da Entidade, o caso de uso retorna para o passo 4 do sub-fluxo 2.3.2 Pesquisar Entidade.

### 2.4.4 Dados Obrigatórios Não Informados

1. Se o usuário não tiver informado os dados obrigatórios em algum dos seguintes passos:
  - Passo 4 do sub-fluxo 2.3.1 Criar Entidade;

- Passo 4 do sub-fluxo 2.3.3 Atualizar Entidade;
- 2. O sistema exibe uma mensagem informando quais dados não foram preenchidos;
- 3. O caso de uso retorna, com os dados do cadastro recém-configurado, aos seguintes passos respectivamente:
  - Passo 4 do sub-fluxo 2.3.1 Criar Entidade;
  - Passo 4 do sub-fluxo 2.3.3 Atualizar Entidade;
- 4. Passo 2 do Fluxo Básico (*Happy Day*).

#### 2.4.5 Entidade Duplicada

1. Se no passo 4 do sub-fluxo 2.3.1 Criar Entidade ou no passo 4 do sub-fluxo 2.3.3 Atualizar Entidade **o sistema encontrar uma Entidade duplicada, o sistema exibe uma mensagem informando que a Entidade já existe;**
2. O caso de uso retorna ao passo 2 do sub-fluxo 2.3.1 Criar Entidade ou no passo 2 do sub-fluxo 2.3.3 Atualizar Entidade, respectivamente, com os dados do cadastro recém-configurado.

#### 2.5 Requisitos Especiais

- Todas os requisitos especiais devem estar contidas no caso de uso específico.

#### 2.6 Pré-Condições

- Todas as pré-condições devem estar contidas no caso de uso específico.

#### 2.7 Pós-Condições

- Todas as pós-condições devem estar contidas no caso de uso específico.

#### 2.8 Relacionamento com Outros Casos de Uso

- Todos os relacionamentos com outros casos de uso devem estar contidas no caso de uso específico.

##### 2.8.1 Pontos de Inclusão

- Todos os pontos de inclusão devem estar detalhados no caso de uso específico.

##### 2.8.2 Pontos de Extensão

- Todos os pontos de extensão devem estar detalhados no caso de uso específico.

# **APÊNDICE C**

## **CASO DE USO RELATÓRIO GENÉRICO**





UNIVERSIDADE  
FEDERAL DO CEARÁ

## ESPECIFICAÇÃO DE CASOS DE USO

**Caso de uso Gerar Relatório**

**Projeto X – Sistema X**

**RESERVADO**

<b>Responsável:</b> Enyo José T. Gonçalves	
<b>Elaborador(es):</b>	<b>e-mail</b>
Micaelly P. Soares	micaellyps@gmail.com

## HISTÓRICO

<b>Data</b>	<b>Versão</b>	<b>Responsável</b>	<b>Alteração</b>
20/10/2012	01	Micaelly P. Soares	- Criação do documento

## ÍNDICE

1	INTRODUÇÃO .....	4
1.1	Objetivos .....	4
1.2	Público Alvo deste Documento .....	4
2	CASO DE USO GERAR RELATÓRIO .....	4
2.1	Descrição .....	4
2.2	Atores Envolvidos.....	4
2.3	Fluxo Básico ( <i>Happy Day</i> ).....	4
2.3.1	Gerar relatório .....	4
2.4	Fluxos Alternativos ( <i>Exceptions</i> ).....	4
2.5	Requisitos Especiais .....	5
2.6	Pré-Condições.....	5
2.7	Pós-Condições .....	5
2.8	Relacionamento com Outros Casos de Uso .....	5
2.8.1	Pontos de Inclusão .....	5
2.8.2	Pontos de Extensão .....	5

# ESPECIFICAÇÃO DE CASOS DE USO

## 1 INTRODUÇÃO

### 1.1 Objetivos

Este documento tem como objetivo detalhar o caso de uso genérico Gerar Relatório.

### 1.2 Público Alvo deste Documento

- Caso tenha algum público alvo específico, estes devem ser detalhados no caso de uso específico Gerar Relatório X.

## 2 CASO DE USO GERAR RELATÓRIO

### 2.1 Descrição

Este caso de uso se responsabiliza pela criação de um relatório genérico.

### 2.2 Atores Envolvidos

- Os atores devem estar detalhados no caso de uso específico Gerar Relatório X.

### 2.3 Fluxo Básico (*Happy Day*)

1. Este caso de uso inicia quando o usuário seleciona a opção Gerar Relatório;
2. O sistema requisita que o usuário especifique a operação que deseja realizar.
3. Uma vez que o usuário solicite executar uma das funções desejadas (X, X), um dos seguintes sub-fluxos é executado:
  - Se o usuário desejar “Criar”, o sub-fluxo 2.3.1 Criar Entidade é executado;

#### 2.3.1 Gerar relatório

1. Este sub-fluxo inicia quando o usuário seleciona a opção Gerar Relatório.
2. O usuário seleciona a opção Nome do Relatório.
3. O sistema exhibe os campos específicos de geração do relatório.
4. O usuário dá entrada com os dados requisitados pelo sistema e solicita a geração do relatório.
5. O relatório é gerado e exibido pelo sistema, mostrando os campos específicos que devem estar contidos no caso de uso específico Gerar Relatório X.

### 2.4 Fluxos Alternativos (*Exceptions*)

Ver se existem fluxos alternativos

## **2.5 Requisitos Especiais**

- Todas os requisitos especiais devem estar contidas no caso de uso específico Gerar Relatório X.

## **2.6 Pré-Condições**

- Todas as pré-condições devem estar contidas no caso de uso específico Gerar Relatório X.

## **2.7 Pós-Condições**

- Todas as pós-condições devem estar contidas no caso de uso específico Gerar Relatório X.

## **2.8 Relacionamento com Outros Casos de Uso**

- Todos os relacionamentos com outros casos de uso devem estar contidas no caso de uso específico Gerar Relatório X.

### **2.8.1 Pontos de Inclusão**

- Todos os pontos de inclusão devem estar detalhados no caso de uso específico Gerar Relatório X.

### **2.8.2 Pontos de Extensão**

- Todos os pontos de extensão devem estar detalhados no caso de uso específico Gerar Relatório X.



# ANEXOS



# **ANEXO A**

## **DOCUMENTO DE VISÃO**





UNIVERSIDADE  
FEDERAL DO CEARÁ

## DOCUMENTO DE VISÃO

<Nome do Projeto>–<Sistema X>

**RESERVADO**

<b>Responsável:</b> Enyo José T. Gonçalves	
<b>Elaborador(es):</b>	<b>e-mail</b>
Micaelly P. Soares	micaellyps@gmail.com

## HISTÓRICO

<b>Data</b>	<b>Versão</b>	<b>Responsável</b>	<b>Alteração</b>
07/04/2013	01	Micaelly P. Soares	- Criação do documento

## ÍNDICE

1	INTRODUÇÃO .....	4
1.1	Objetivos .....	4
1.2	Público Alvo deste Documento .....	4
2	POSICIONAMENTO .....	4
2.1	Descrição do Problema .....	4
2.2	Sentença de Posição do Produto .....	4
3	DESCRIÇÃO DOS ENVOLVIDOS E DOS USUÁRIOS .....	5
3.1	Resumo dos Envolvidos .....	5
3.2	Resumo dos Usuários .....	5
3.3	Ambiente do Usuário .....	6
3.4	Principais Necessidades dos Usuários ou dos Envolvidos .....	6
3.5	Alternativas e Concorrência .....	6
4	VISÃO GERAL DO PRODUTO .....	6
4.1	Perspectiva do Produto .....	6
4.2	Suposições e Dependências .....	7
5	RECURSOS DO PRODUTO .....	7
6	OUTROS REQUISITOS DO PRODUTO .....	7

# DOCUMENTO DE VISÃO

## 1 INTRODUÇÃO

*[A finalidade deste documento é coletar, analisar e definir necessidades e recursos de nível superior do <<Nome do Sistema>>. Ele se concentra nos recursos necessários aos envolvidos e aos usuários-alvo e nas razões que levam a essas necessidades. Os detalhes de como o <<Nome do Sistema>> satisfaz essas necessidades são descritos no caso de uso e nas especificações suplementares.]*

*[A introdução do documento Visão fornece uma visão geral de todo o seu conteúdo. Ela contém a finalidade e as referências desse documento.]*

### 1.1 Objetivos

*[Forneça uma descrição resumida dos objetivos deste documento.]*

### 1.2 Público Alvo deste Documento

*[Esta subseção fornece uma lista completa do público alvo deste documento.]*

## 2 POSICIONAMENTO

### 2.1 Descrição do Problema

*[Forneça uma descrição resumindo o problema que está sendo resolvido pelo projeto. Poderá ser usado este formato:]*

<b>O problema de</b>	<i>[descreva o problema]</i>
<b>afeta</b>	<i>[os envolvidos afetados pelo problema]</i>
<b>cujo impacto é</b>	<i>[qual é o impacto do problema?]</i>
<b>uma boa solução seria</b>	<i>[liste alguns dos principais benefícios de uma boa solução]</i>

### 2.2 Sentença de Posição do Produto

*[Forneça uma sentença geral resumindo, no nível mais alto, a posição exclusiva que o produto pretende ocupar no mercado. Poderá ser usado este formato:]*

<b>Para</b>	<i>[cliente-alvo]</i>
<b>Que</b>	<i>[indique a necessidade ou oportunidade]</i>
<b>O (nome do produto)</b>	<i>é um(a) [categoria do produto]</i>
<b>Que</b>	<i>[indique o principal benefício; ou seja, a razão convincente que motiva a compra]</i>
<b>Ao contrário de</b>	<i>[principal alternativa da concorrência]</i>
<b>Nosso produto</b>	<i>[indique a principal diferença]</i>

*[Uma sentença de posição do produto comunica o objetivo do aplicativo e a importância do projeto para todo o pessoal envolvido.]*

### 3 DESCRIÇÃO DOS ENVOLVIDOS E DOS USUÁRIOS

*[Para fornecer, de maneira eficiente, produtos e serviços que atendam às reais necessidades dos usuários e envolvidos, é necessário identificar e considerar todos os envolvidos como parte do processo de Modelagem de Requisitos. É necessário também identificar os usuários do sistema e assegurar que a comunidade de envolvidos os represente adequadamente. Esta seção fornece um perfil dos envolvidos e dos usuários que integram o projeto, e dos principais problemas que, de acordo com o ponto de vista deles, poderão ser abordados pela solução proposta. Ela não descreve as solicitações ou os requisitos específicos dos usuários e dos envolvidos, já que eles são capturados em um artefato individual de solicitações dos envolvidos. Em vez disso, ela fornece a base e a justificativa que explicam por que os requisitos são necessários.]*

#### 3.1 Resumo dos Envolvidos

*[Há uma série de envolvidos que se interessam pelo desenvolvimento e nem todos eles são usuários finais. Apresente uma lista resumida desses envolvidos que não são usuários. (O resumo dos usuários encontra-se na seção 3.2.)]*

Nome	Descrição	Responsabilidades
<i>[Especifique o nome do tipo de envolvido.]</i>	<i>[Descreva brevemente o envolvido.]</i>	<p><i>[Resuma as principais responsabilidades do envolvido no que diz respeito ao sistema que está sendo desenvolvido; ou seja, seu interesse como envolvido. Por exemplo, este envolvido:</i></p> <ul style="list-style-type: none"> <li><i>- assegura que o sistema poderá ser mantido</i></li> <li><i>- assegura que haverá uma demanda de mercado pelos recursos do produto</i></li> <li><i>- monitora o andamento do projeto</i></li> <li><i>- aprova financiamentos</i></li> <li><i>- e assim por diante]</i></li> </ul>

#### 3.2 Resumo dos Usuários

*[Apresente uma lista resumida de todos os usuários identificados.]*

Nome	Descrição	Responsabilidades	Envolvido
<i>[Informe o tipo de usuário.]</i>	<i>[Descreva brevemente o que ele representa no que diz respeito ao sistema.]</i>	<p><i>[Liste as principais responsabilidades do usuário em relação ao sistema que está sendo desenvolvido; por exemplo:</i></p> <ul style="list-style-type: none"> <li><i>- percebe os detalhes</i></li> <li><i>- elabora relatórios</i></li> <li><i>- coordena o trabalho</i></li> </ul>	<i>[Se o usuário não for representado diretamente, identifique o envolvido responsável por representar os interesses dele.]</i>

### 3.3 Ambiente do Usuário

*[Descreva o ambiente de trabalho do usuário-alvo. A seguir são apresentadas algumas sugestões:*

- *Número de pessoas envolvidas na execução da tarefa? Isso está mudando?*
- *Qual é a duração de um ciclo de tarefas? Qual é o tempo gasto em cada atividade? Isso está mudando?*
- *Quaisquer restrições ambientais exclusivas: móveis, externas, de aeronaves etc?*
- *Que plataformas de sistema são utilizadas hoje? Plataformas futuras?*
- *Que outros aplicativos estão em uso? É necessário que o seu aplicativo interaja com eles?*

*Nesse ponto, você poderá incluir textos provenientes do Modelo de Negócios para descrever a tarefa e os trabalhadores de negócio envolvidos, entre outros.]*

### 3.4 Principais Necessidades dos Usuários ou dos Envolvidos

*[Liste os principais problemas com as soluções existentes conforme o ponto de vista do envolvido ou do usuário. Esclareça as seguintes questões referentes a cada problema:*

- *Quais são as causas deste problema?*
- *Como ele está sendo resolvido agora?*
- *Que soluções o envolvido ou o usuário deseja?]*

*[É importante compreender a importância relativa exercida pelo usuário ou pelo envolvido na resolução de cada problema. As técnicas de ordenação e votação cumulativa indicam os problemas que devem ser resolvidos versus problemas que eles gostariam que fossem resolvidos.]*

Necessidade	Prioridade	Preocupações	Solução Atual	Soluções Propostas
Mensagens de difusão				

### 3.5 Alternativas e Concorrência

*[Identifique as alternativas que o envolvido considera disponíveis. Entre elas podem estar incluídas a compra de um produto do concorrente, a criação de uma solução local ou a simples manutenção do status quo. Liste todas as opções conhecidas que a concorrência oferece ou que podem se tornar disponíveis. Inclua os principais pontos fortes e pontos fracos de cada concorrente segundo o ponto de vista do envolvido ou do usuário final.]*

## 4. Visão Geral do Produto

### 4 VISÃO GERAL DO PRODUTO

*[Esta seção fornece uma visão de nível superior dos recursos, interfaces com outros aplicativos e configurações de sistemas do produto. Ela geralmente é constituída destas duas subseções:*

- *Perspectiva do produto*
- *Suposições e dependências]*

#### 4.1 Perspectiva do Produto

*[Esta subseção do documento Visão analisa o produto em relação a outros produtos relacionados e ao ambiente do usuário. Se o produto for independente e totalmente auto-suficiente, exponha isso aqui. Se o produto for um componente de um sistema maior, esta subseção relatará como esses sistemas interagem e terá de identificar as interfaces relevantes entre os sistemas. Uma maneira fácil de exibir os principais componentes do sistema maior, suas interconexões e interfaces externas é através de um diagrama de*

*bloco.]*

## 4.2 Suposições e Dependências

*[Liste cada fator que afeta os recursos especificados no documento Visão. Liste as suposições que, se sofrerem mudanças, alterarão o documento Visão. Por exemplo, uma suposição poderá estabelecer que um sistema operacional específico estará disponível para o hardware projetado para o produto de software. Se o sistema operacional não estiver disponível, o documento Visão terá que ser alterado.]*

## 5 RECURSOS DO PRODUTO

*[Liste e descreva brevemente os recursos do produto. Trata-se dos recursos de nível superior do sistema que são necessários para propiciar benefícios aos usuários. Cada recurso é um serviço desejado externamente que normalmente exige uma série de entradas para alcançar os resultados desejados. Por exemplo, um dos recursos de um sistema de rastreamento de problemas poderá ser a capacidade de fornecer relatórios de tendências. À medida que o modelo de casos de uso for desenvolvido, atualize a descrição para fazer referência aos casos de uso.*

*Como o documento Visão é revisado por uma ampla variedade de pessoas envolvidas, o nível de detalhamento terá que ser genérico o bastante para que todos possam compreendê-lo. No entanto, devem estar disponíveis detalhes suficientes para fornecer à equipe as informações necessárias para criar um modelo de casos de uso.*

*Para gerenciar a complexidade dos aplicativos de maneira eficiente, é recomendável para qualquer sistema novo, ou para uma adição que complemente um sistema existente, que seja utilizado um grau de abstração de nível suficientemente elevado de modo a resultar em 25 a 99 recursos. Esses recursos serão a base fundamental do gerenciamento do projeto, do gerenciamento do escopo e da definição do produto. Cada recurso será descrito mais detalhadamente no modelo de casos de uso.*

*Em toda esta seção, cada recurso poderá ser externamente percebido por usuários, operadores e outros sistemas externos. Esses recursos deverão incluir uma descrição da funcionalidade e de todas as questões de usabilidade relevantes que deverão ser abordadas. As seguintes diretrizes se aplicam:*

- Evite o design. Mantenha as descrições dos recursos em um nível geral. Concentre-se nos recursos necessários e no porquê (e não em como) eles deverão ser implementados.*
- Se estiver usando o kit de ferramentas do Rational RequisitePro, tudo terá que ser selecionado como requisitos de tipo para facilitar a consulta e o rastreamento.*

*Defina a prioridade dos diferentes recursos do sistema. Inclua, se for útil, atributos como, por exemplo, estabilidade, benefício, esforço e risco.]*

## 6 OUTROS REQUISITOS DO PRODUTO

*[Em um nível superior, liste padrões aplicáveis, requisitos de hardware ou de plataforma; requisitos de desempenho; e requisitos ambientais.*

*Defina as faixas de qualidade para desempenho, robustez, tolerância a erros, usabilidade e características semelhantes que não são capturadas no Conjunto de Recursos.*

*Observe quaisquer restrições de design, restrições externas ou outras dependências.*

*Defina quaisquer requisitos de documentação específicos, incluindo requisitos de manuais do usuário, Ajuda on-line, instalação, rotulagem e de embalagem.*

*Defina a prioridade desses outros requisitos do produto. Inclua, se for útil, atributos como, por exemplo, estabilidade, benefício, esforço e risco.]*



**ANEXO B**  
**DOCUMENTO DE ESPECIFICAÇÃO**  
**DOS REQUISITOS**





UNIVERSIDADE  
FEDERAL DO CEARÁ

## ESPECIFICAÇÃO DE REQUISITOS

<Nome do Projeto> – <Sistema X>

**RESERVADO**

<b>Responsável:</b> Enyo José T. Gonçalves	
<b>Elaborador(es):</b>	<b>e-mail</b>
Carla Ilane Moreira Bezerra	carlailane@gmail.com

## HISTÓRICO

<b>Data</b>	<b>Versão</b>	<b>Responsável</b>	<b>Alteração</b>
11/04/2013	01	Micaelly P. Soares	- Criação do documento

## ÍNDICE

1	INTRODUÇÃO .....	4
1.1	Objetivos .....	4
1.2	Público alvo deste documento .....	4
1.3	Glossário .....	4
1.4	Referências .....	4
2	REQUISITOS DO SISTEMA .....	4
2.1	Requisitos Funcionais .....	4
2.2	Regras de Negócio.....	4
2.3	Requisitos Não Funcionais.....	5

## ESPECIFICAÇÃO DE REQUISITOS

### 1 INTRODUÇÃO

#### 1.1 Objetivos

*[Inserir objetivo do documento]*

#### 1.2 Público alvo deste documento

*[Inserir público alvo]*

#### 1.3 Glossário

Verbetes	Definição

#### 1.4 Referências

[1]

### 2 REQUISITOS DO SISTEMA

*[Quaisquer funções, restrições ou propriedade que o sistema realizar, obedecer ou satisfazer de forma a realizar o que seus usuários desejam]*

#### 2.1 Requisitos Funcionais

Id	Interessado	Descrição

#### 2.2 Regras de Negócio

Id	Interessado	Descrição

### 2.3 Requisitos Não Funcionais

Id	Interessado	Descrição

