



UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
BACHARELADO EM ENGENHARIA DE SOFTWARE

**LAISA MORAIS DA ROCHA**

**APLICAÇÃO DE UM *SUBSET* DE PRÁTICAS DO TMMI EM NÚCLEO  
DE PRÁTICAS EM INFORMÁTICA DA UFC CAMPUS QUIXADÁ**

**QUIXADÁ  
2014**

**LAISA MORAIS DA ROCHA**

**APLICAÇÃO DE UM *SUBSET* DE PRÁTICAS DO TMMI EM NÚCLEO  
DE PRÁTICAS EM INFORMÁTICA DA UFC CAMPUS QUIXADÁ**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Orientador Prof. Carlos Diego Andrade de Almeida.

**QUIXADÁ  
2014**

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

---

R573a Rocha, Laisa Morais da  
Aplicação de um *subset* de práticas do TMMi em núcleo de práticas em informática da UFC  
Campus Quixadá / Laisa Morais da Rocha. – 2014.  
49 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de  
Engenharia de Software, Quixadá, 2014.

Orientação: Prof. Me. Carlos Diego Andrade de Almeida  
Área de concentração: Computação

1. Software - Testes 2. Software de aplicação 3. Software – Qualidade do projeto  
I. Título.

---

CDD 005.30287

**LAISA MORAIS DA ROCHA**

**APLICAÇÃO DE UM *SUBSET* DE PRÁTICAS DO TMMI EM NÚCLEO  
DE PRÁTICAS EM INFORMÁTICA DA UFC CAMPUS QUIXADÁ**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Aprovado em: 27 / novembro / 2014.

**BANCA EXAMINADORA**

---

Prof. Msc. Carlos Diego Andrade de Almeida  
Universidade Federal do Ceará (UFC)

---

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

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Paulyne Mattews Jucá  
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha família e amigos.

## **AGRADECIMENTOS**

Ao meu orientador Prof. Msc. Diego Andrade, pelo apoio, paciência e o incentivo à busca constante pelo conhecimento.

A meus amigos, pela cobrança assídua e construtiva que me ajudou a chegar até aqui.

À minha família, que mesmo longe nunca deixou de acreditar em mim.

E a todos que direta ou indiretamente fizeram parte desta jornada comigo, o meu muito obrigada.

“A tarefa não é tanto ver aquilo que ninguém viu,  
mas pensar o que ninguém ainda pensou  
sobre aquilo que todo mundo vê.”  
(Arthur Schopenhauer)

## RESUMO

Empresas de pequeno porte encontram diversas dificuldades para implementar ou melhorar seus processos de teste devido a falta de profissionais especializados em teste, falta de recursos e a dificuldade em adaptar modelos de maturidade em teste para o contexto específico da empresa. O objetivo deste trabalho é aplicar um conjunto de práticas consideradas essenciais pela comunidade de testes no processo de teste do Núcleo de Práticas em Informática da Universidade Federal do Ceará. Este trabalho apresenta a estratégia utilizada para aplicação, desde o diagnóstico do processo em uso à elaboração e aplicação da proposta de processo. O resultado deste trabalho é uma proposta de um processo de teste enxuto, porém completo e de acordo com as necessidades da empresa, bem como o relato das dificuldades de aplicação e da melhoria do processo de teste da empresa. A proposta do processo leva em consideração o modelo de maturidade em testes TMMi (*Test Maturity Model Integration*), sendo portanto uma proposta com foco em qualidade.

Palavras chave: Teste de software. Melhoria de processo. TMMi. Definição de processo. Empresas de pequeno porte.



## **ABSTRACT**

Small companies face several barriers to implement their software testing processes due to the lack of experts in testing, the lack of resources and difficulties to adapt testing-related maturity models to a particular context of the company. The goal of this work is to apply a set of practices considered essential by test community in the testing process of Nucleus Practices in the Federal University of Ceará. This paper presents the strategy used for the application, from diagnosis of the process in use for developing and implementing the process of proposal. The result of this work is a proposal for a streamlined testing process, but complete and in accordance with the company's needs, as well the report of the implementation difficulties and improving the company's testing process. The testing process takes into consideration the TMMi (Test Maturity Model Integration) model, thus resulting in a process whose focus is on quality.

**Keywords:** Software testing. Process improvement. TMMi. Process definition. Small companies.

## **LISTA DE ILUSTRAÇÕES**

Figura 1 – Níveis de maturidade e áreas de processo do TMMi.....	21
Figura 2 – Processo de teste formal do NPI-UFC.....	30
Figura 3 – Mapa mental da proposta de processo de teste para o NPI.....	34

## LISTA DE TABELAS

Tabela 1 – Aderência do processo real ao <i>subset</i> .....	32
--	----

## SUMÁRIO

1 INTRODUÇÃO.....	16
2 FUNDAMENTAÇÃO TEÓRICA .....	16
2.1 Processo de Software .....	17
2.2 Qualidade de Software .....	17
2.3 Teste de Software .....	18
2.4 Modelos de Maturidade .....	19
2.4.1 CMMI .....	19
2.4.2 MPS.Br .....	20
2.4.3 MPT.Br .....	20
2.4.4 Test Maturity Model Integration – TMMi.....	21
2.5 <i>Subset</i> TMMi .....	23
3 TRABALHOS RELACIONADOS .....	26
4 PROCEDIMENTOS.....	27
4.1 AMBIENTE EM ESTUDO .....	28
4.1.1 Perfil do NPI.....	28
4.1.2 Estrutura organizacional.....	28
4.2 ESTABELECIMENTO DO PROCESSO A SER APLICADO.....	29
4.2.1 Definição de estratégia .....	29
4.2.2 Diagnóstico do processo de testes atual .....	29
4.2.3 Processo real.....	31
4.2.4 Análise da aderência do processo real ao <i>subset</i> do TMMi .....	31
4.2.5 Aplicação do modelo genérico .....	33
4.3 Implementação do modelo genérico .....	38
5 DESENVOLVIMENTO.....	38
5.1 Projeto-piloto GPA (Gestão de Programas Acadêmicos).....	38
5.2 Projeto GAL (Gestão de Aquisição de Livros).....	40
6 DISCUSSÃO .....	41
6.1 Dificuldades .....	42
6.2 Lições Aprendidas .....	42
6.3 Possíveis melhorias.....	43
7 CONSIDERAÇÕES FINAIS .....	43
7.1 Contribuições .....	44
7.2 Limitações.....	44
7.3 Trabalhos Futuros .....	45
REFERÊNCIAS .....	46
ANEXOS .....	48
ANEXO A – Plano de Teste.....	48
ANEXO B – <i>Print-screen</i> da Planilha de Teste do NPI.....	51

## 1 INTRODUÇÃO

A tecnologia da informação tem deixado de ser uma mera ferramenta de suporte e tornou-se parte ativa no mercado, com o dever de impulsionar as amplas capacidades que as tecnologias de software e internet oferecem para que se possa competir e obter destaque no mercado. Deste modo, as empresas de software têm buscado continuamente melhorar, investir na excelência operacional e reduzir custos em geral e custos fixos, para que se possam concentrar energias em busca de crescimento, inovação e transformação.

Atualmente, essas empresas têm investido fortemente nos sistemas de informação e, sabendo da influência que a área de TI exerce atualmente, é imprescindível que a confiabilidade do software seja garantida. Conseqüentemente, as atividades que garantam a qualidade do produto devem estar presentes durante todo o processo de desenvolvimento do software. Neste trabalho, priorizaremos a fase de teste de software, que visa garantir um produto com menor quantidade de defeitos e, portanto, mais confiável.

No entanto, pequenas empresas tem apresentado dificuldades para implementar um processo de teste. Rodrigues, Pinheiro e Albuquerque (2010) apontam diversas causas para o problema, mas enfatizam que a falta de profissionais especializados em teste; falta de recursos; falta de conhecimento para executar adequadamente as atividades de teste e dificuldade em adaptar modelos de maturidade em teste para o contexto específico da empresa são os principais motivos que dificultam a implementação de um processo. É necessário, à vista disso, estabelecer um processo de teste bem definido e adequá-lo à realidade da empresa, o que não é algo trivial devido à dificuldade de adaptação dos modelos de referência existentes.

Segundo o TMMi (TMMi, 2010), nos últimos anos a indústria de software tem investido grandes esforços para melhorar a qualidade de seus produtos. O que tem sido uma tarefa difícil, já que o tamanho e a complexidade do software aumentam rapidamente, paralelamente aos clientes e usuários que estão cada vez mais exigentes. A fim de aperfeiçoar a qualidade do produto, a indústria de software tem focado em melhorar seus processos de desenvolvimento. É considerado também que, apesar de pelo menos 30-40% dos custos totais do projeto serem de responsabilidade da fase de testes, pouca atenção é voltada a esta nos vários modelos de melhoria de processo de software, incluindo o CMM e o CMMI. Tendo consciência disso, a resposta que a comunidade de testes encontrou fora criar os seus próprios modelos de melhoria, dentre eles o TMMi (Modelo de Maturidade para Testes Integrado).

É importante ressaltar que esses modelos são caros. O TMMi, por exemplo, não possui instituição avaliadora e implementadora no Brasil, o que implica o envolvimento de instituições implementadoras estrangeiras, encarecendo ainda mais o processo de certificação nesse modelo. Além disso, muito se é sugerido sobre o que deve ser feito, mas não é abordado como se fazer e priorizar essas atividades. Esses modelos não oferecem à empresa um meio de elaborar seu processo sozinha, pois são complexos e de difícil compreensão, já que exigem muito esforço para sua interpretação. Para pequenas empresas, por exemplo, pode ser inviável a utilização das práticas sugeridas, uma vez que estas podem não se aplicar a realidade da empresa. Nesses casos, o processo de teste elaborado deve contemplar todas as fases do teste de software, mas com a quantidade de atividades de acordo com a realidade da organização e o modelo de desenvolvimento de software utilizado. Se deve, portanto, tornar o processo de teste mais simples de ser executado, não exigindo muitos recursos, nem equipe muito grande, facilitando o alcance do objetivo de melhorar a qualidade dos produtos.

Camargo (2013) identifica um *subset* (subconjunto) de práticas do TMMi para estabelecer um processo de teste de software simplificado para pequenas e médias empresas. Seu artigo descreve um survey aplicado a profissionais de teste de software, estabelecendo um processo genérico que apresenta as práticas essenciais de um processo bem definido, reduzindo em cerca de 60% o número de técnicas utilizadas no TMMi. O processo genérico justifica-se uma vez que nem todas as práticas do TMMi são viáveis para todos os tamanhos de empresas e equipes. Assim, é importante estar ciente de um conjunto básico de práticas que, se não realizadas, podem comprometer a qualidade do processo e, conseqüentemente, a qualidade do produto testado.

O objetivo principal deste trabalho é relatar a aplicação de um *subset* de práticas do TMMi através de um estudo de caso real, no Núcleo de Práticas de Informática da Universidade Federal do Ceará, no campus de Quixadá. Será feita uma comparação do processo real do NPI com o *subset* a fim de gerar uma proposta de processo de teste simplificado condizente com a realidade no NPI. Em seguida, será feita uma análise da realização das atividades do processo simplificado em um projeto piloto do NPI, a fim de identificar e relatar dificuldades e melhorias da proposta.

## **2 FUNDAMENTAÇÃO TEÓRICA**

A seguir são apresentados em mais detalhes os conceitos-chave deste trabalho. Nas primeiras duas subseções serão explanados os conceitos de processo e qualidade de software, respectivamente. Na subseção seguinte é apresentado o conceito de teste de

software, bem como algumas observações sobre processo de teste. Por fim, os modelos de maturidade CMMI, MPS.br, MPT.br e TMMi são brevemente apresentados, com maior ênfase e detalhamento no TMMi, uma vez que é o modelo que será utilizado como base para este trabalho. O *subset* será apresentado logo em seguida, por ser o conjunto de práticas que será adaptado à realidade da empresa onde o estudo de caso será executado.

## 2.1 Processo de Software

Não há um consenso no que diz respeito ao conceito de Processo de Software, uma vez que as abordagens têm variado ao longo dos anos. Focando no desenvolvimento de software, Sommerville (2007) define um processo de software como um conjunto de atividades que leva à produção de um produto de software. Humphrey (1989) afirma que para desenvolver um produto de software é necessário seguir passos ou atividades bem definidos para que possam ser repetidos. O conjunto dessas atividades, quando é devidamente executado e produz o resultado esperado, é chamado de processo.

A abordagem mais coerente com este trabalho é a de PRESSMAN (2006), que define processo de software como um arcabouço para as tarefas que são necessárias para construir softwares de alta qualidade. É também a abordagem que será adotada, uma vez que é compatível com uma das finalidades deste trabalho após a aplicação do processo genérico.

Ao se ter um processo de software voltado exclusivamente para testes, se espera que as atividades sejam executadas sempre da mesma maneira, resultando em um produto final de maior qualidade. Para isso, é necessário que os passos requeridos estejam bem definidos de forma que a atividade possa ser repetida sempre que preciso, o que caracteriza um processo. Portanto, para que o teste de software possa ser realizado com qualidade é necessário que um processo esteja definido.

## 2.2 Qualidade de Software

Hipoteticamente intuitivo, o conceito de qualidade é simples; contudo, ao refletir um pouco mais sobre a idéia de qualidade, percebe-se que o conceito deve ser reconsiderado. Segundo Weinberg (2007), qualidade é relativa, uma vez que o que é qualidade pra uns pode ser falta de qualidade para outros.

É comum que a definição de qualidade esteja associada à inexistência de *bugs* em um produto, mas é possível encontrar softwares defeituosos com usuários bastante satisfeitos. É preciso entender até que ponto os *bugs* influenciam na qualidade de um produto.

A qualidade de um software está relacionada à entrega ao cliente de um produto final que satisfaça suas expectativas, conforme acordado inicialmente por meio dos requisitos do projeto. Dessa forma, a qualidade de software é uma área da engenharia de software que tem como objetivo garantir a qualidade pela definição de processos de desenvolvimento (ENGHOLM, 2010). Abordando desta forma, chegamos à definição de CROSBY (1992): "A qualidade é conformidade aos requisitos". Considerando que requisitos são as necessidades que foram expressas pelos clientes, é interessante perceber que há um pensamento embutido neste conceito – de que é preciso um ponto de referência para que se possa julgar um produto.

De acordo com Koscianski (2007), a maioria das pessoas, incluindo estudantes de TI, assume que qualidade de software e *bugs* são coisas incompatíveis, o que não é verdade.

A qualidade possui alguns princípios básicos, como:

- Tentar prevenir defeitos ao invés de consertá-los;
- Ter a certeza de que os defeitos que foram encontrados sejam corrigidos o mais rápido possível;
- Estabelecer e eliminar as causas, bem como os sintomas dos defeitos;
- Auditar o trabalho de acordo com padrões e processos previamente estabelecidos.

Muitas empresas de software vem adotando processos de teste como forma de garantir uma maior qualidade de seus produtos e uma posição de destaque no mercado de trabalho. A qualidade não é um atributo que pode ser imposto ao produto depois de finalizado (JINO; MALDONADO; DELAMARO, 2007), o que ressalta a importância de começar a testar o software desde o início do desenvolvimento. Veremos um pouco mais sobre teste de software na subseção a seguir.

### **2.3 Teste de Software**

Segundo Herbert (1999), teste de software é formado por um conjunto de atividades de validação e verificação que tem como objetivo encontrar defeitos no produto e no processo de software. Myers (2004), por sua vez, define teste como o ato de executar um programa fornecendo dados de entrada, conferir os resultados obtidos e compará-los com os resultados esperados. Ele ainda completa: “Testar um software consiste em executá-lo na tentativa de revelar o maior número possível de defeitos”, buscando aumentar a qualidade do produto em questão.

De acordo com o Instituto de Engenharia de Software (SEI), órgão criado pelo Departamento de Defesa americano, cerca de 55% de todo trabalho realizado pela área de



desenvolvimento de software de uma empresa é destinado à correção de erros que foram detectados em sistemas já finalizados. Defeitos podem ser introduzidos em qualquer fase do desenvolvimento de software, e o quanto antes forem identificados, menor será o custo de sua correção (MALDONADO; FABBRI, 2001). Sendo assim, é imprescindível que haja um investimento no que diz respeito a definição e implantação de processos de teste que permitam o desenvolvimento de softwares mais confiáveis e com menores custos.

O processo de teste visa então, apoiar o desenvolvimento com o intuito de que o produto final esteja de acordo com o que fora especificado. O sucesso do desenvolvimento de software e da atividade de teste dependem diretamente de um bom planejamento e da qualidade dos processos utilizados. Um processo de teste, assim como de software, possui uma série de fases, procedimentos e passos (BURNSTEIN, 2003), então é importante que seja bem planejado.

## **2.4 Modelos de Maturidade**

O conceito básico sob o termo maturidade é de que as organizações maduras fazem as coisas de modo sistemático (SIQUEIRA, 2005). Um modelo de maturidade funciona como um guia para a organização, de tal maneira que ela possa localizar onde está e como está, “espelhando-se” nele para que, em seguida, se possa realizar um plano para alcançar um nível melhor que o atual, na busca pela excelência.

Nas subseções a seguir serão apresentados brevemente os modelos de melhoria considerados importantes para o melhor entendimento deste trabalho.

### **2.4.1 CMMI**

As empresas podem definir seus processos baseando-se em modelos de referência. Como dito anteriormente, modelos de referência indicam as melhores práticas que podem ser utilizadas na elaboração do processo, bem como fornecem diretrizes do que deve ser implementado. Não sugerem, entretanto, o modo como este processo será implementado na organização.

Estes modelos, além de auxiliar na elaboração de processos, também permitem que as organizações obtenham maturidade para analisar o processo que já está em uso buscando por melhorias. Desenvolvido em 1997, pela *Carnegie Mellon University* em parceria com a *Systems Engineering Institute* - SEI, o *Capability Maturity Model* (CMM), foi o primeiro modelo de maturidade criado. Ele oferece um meio de avaliar o processo de desenvolvimento de software fornecendo diretrizes de melhores práticas que podem ser adotadas (SEI, 2006). O

CMMI (SEI, 2006) é uma evolução do CMM e objetiva ajudar organizações a melhorar o desenvolvimento e a manutenção de processos de produtos e serviços.

#### **2.4.2 MPS.Br**

O modelo MPS.BR surgiu em dezembro de 2003 como necessidade de melhoria de processo do software brasileiro (WEBER, et al., 2006). Ele define um modelo de referência para melhoria e avaliação de processos de software de forma a atender as necessidades de negócio de empresas brasileiras (SOFTEX, 2011).

O MPS.Br oferece uma maneira mais barata, comparada com padrões internacionais como o CMMI, visto anteriormente, para que uma pequena ou média empresa adquira maturidade em seus processos de desenvolvimento de software, tornando-se, dessa forma, mais competitiva. O MPS.Br foca o desenvolvimento de software em geral, e inclui a área de teste, assim como o CMMI (SEI, 2006).

O modelo de referência do MPS.Br está organizado em guias de implementação que indicam o que deve ser implementado, além de um guia geral que descreve o modelo. Uma vez que é crescente o número de empresas que fornecem serviços exclusivamente em teste de software, o MPS.Br oferece um guia de implementação específico para organizações do tipo fábrica de teste (MPS.BR, 2011). Além disso, foi criado um modelo exclusivamente para processo de teste, o Melhoria do Processo de Teste Brasileiro (MPT.Br) (MPT.BR, 2011) que veremos com mais detalhes na próxima subseção.

#### **2.4.3 MPT.Br**

O MPT.Br é um modelo de Melhoria do Processo de Teste concebido para apoiar organizações através dos elementos essenciais para o desenvolvimento da disciplina de teste, inserida no processo de desenvolvimento de software. O MPT.Br trata a melhoria do processo de teste através das melhores práticas relativas às atividades desenvolvidas ao longo do ciclo de vida de teste do produto.

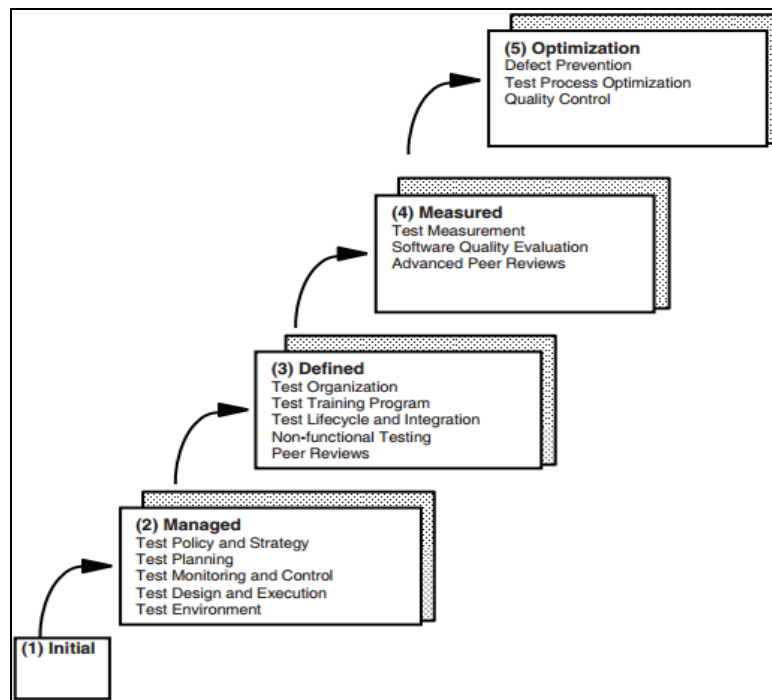
O MPT.Br foi elaborado e é mantido pelo Softex Recife e RioSoft. Para a construção do MPT.Br, tomou-se como base diversos modelos de maturidade em teste de software, entre eles o CMMI, MPS.Br e o próprio TMMi. O objetivo da construção do MPT.Br foi criar uma certificação alternativa brasileira e que não envolvesse instituições estrangeiras, como é o caso do TMMi que veremos adiante. Conseqüentemente, o custo seria reduzido e mais empresas brasileiras de diversos portes poderiam implantar e melhorar seu processo de teste. O MPT.Br objetiva a melhoria do processo de teste por meio da realização das melhores práticas ao

longo de todo ciclo de vida de teste do produto (MPT.BR, 2011) e pode ser visto como complementar ao MPS.Br. Entretanto, trata-se de um modelo ainda pouco conhecido, sendo que as primeiras empresas foram certificadas no ano de 2010 e, de acordo com o portal do MPT.Br, apenas uma empresa no Brasil é certificada no nível 5 (nível máximo) até agora.

#### 2.4.4 Test Maturity Model Integration – TMMi

Complementar ao CMMI, o TMMi (TMMi Foundation, 2010) foca o esforço de melhoria no processo de teste. O TMMi foi desenvolvido pela TMMi Foundation como um guia e um framework para a melhoria de processos de teste. O TMMi é semelhante ao CMMI em sua estrutura, pois possui níveis de maturidade para avaliação e melhoria de processo, como mostra a Figura 1.

Figura - 1 Níveis de maturidade e áreas de processo do TMMi



Fonte: TMMi (2014)

Com exceção do nível 1, cada nível de maturidade é composto de algumas áreas de processos que indicam onde uma organização deve focar para melhorar o seu processo de teste e, conseqüentemente, atingir o nível de maturidade desejado. Áreas de processos identificam as questões que devem ser satisfeitas para que se possa alcançar determinado nível de maturidade. Vale ressaltar que, para cada nível, todas as áreas de processo devem ser satisfeitas e cada área de processo identifica um conjunto de atividades de teste que devem ser realizadas. A seguir vemos um detalhamento maior sobre os níveis do TMMi, conforme a figura anterior.

**Nível 1 – Inicial:** Neste nível o processo de teste é caótico, uma vez que não fora definido. O objetivo dos testes aqui é de mostrar que o software funciona sem maiores falhas. Organizações que se encontram neste nível de maturidade caracterizam-se pela tendência ao abandono dos processos em momentos de crise, e também pela incapacidade de repetir seus casos de sucesso.

Nenhuma área de processo é identificada nesse nível.

**Nível 2 – Gerenciado:** Neste nível o processo é definido e ajuda a garantir que as práticas existentes são mantidas durante períodos de crise. O principal objetivo do teste em uma organização de nível 2 é verificar se o produto satisfaz os requisitos especificados. Porém, o teste ainda é percebido como sendo uma fase do projeto que deve ocorrer após a codificação, o que acarreta muitos problemas de qualidade a este nível.

As áreas de processo no TMMi nível 2 são:

- Política de Teste e Estratégia
- Planejamento de Teste
- Acompanhamento e Controle de Teste
- Teste de Projeto e Execução
- Ambiente de Teste

**Nível 3 – Definido:** Neste nível o teste não é mais confinado a uma fase que segue a codificação, ele está integrado no ciclo de vida do software como um todo. Organizações de nível 3 compreendem a importância de avaliações no controle de qualidade; um programa formal de avaliação é implementado, apesar de ainda não ser totalmente ligado ao processo de teste dinâmico. A distinção fundamental entre este e o nível 2 é o escopo dos padrões, descrições de processos, e procedimentos.

As áreas de processo a nível TMMi 3 são:

- Teste de Organização
- Programa de Treinamento de Teste
- Teste do Ciclo de Vida e Integração
- Testes Não-Funcionais
- Revisão por Pares

**Nível 4 – Mensurado:** Os testes são completamente definidos, bem fundamentados e medidos. Revisões e inspeções são incorporadas ao ciclo de vida do desenvolvimento e consideradas parte dos testes. Os produtos de software são avaliados a partir de critérios de qualidade, como reusabilidade, usabilidade e manutenibilidade.

As áreas de processo no TMMi nível 4 são:

- Medição de Teste
- Avaliação da Qualidade do Produto
- Revisão por Pares Avançada

**Nível 5 – Otimizado:** A realização de todas as metas de melhoria nos níveis 1 a 4 criam uma infra-estrutura organizacional para teste que suporta um processo completamente definido e medido. Uma organização nível 5, por sua vez, é capaz de melhorar continuamente seus processos com base em uma compreensão quantitativa dos processos controlados estatisticamente. No nível 5, métodos e técnicas são otimizados e estão em melhoramento contínuo. As três áreas de processo do nível 5 fornecer suporte para essa melhoria contínua.

As áreas de processo no nível TMMi 5 são:

- Prevenção de Defeitos
- Controle de Qualidade
- Processo de Teste de Otimização

Como pode ser visto, o TMMi possui uma estrutura que indica claramente o que deve ser feito para que determinado nível possa ser atingido, porém não indica quais áreas de processo são mais importantes ou que poderiam ser implantadas primeiro em casos de implementações parciais. A implementação deve começar sempre a partir do nível mais baixo que a empresa se encontra e evoluir conforme seu processo ganha maturidade, entretanto, mesmo uma empresa de pequeno porte, por exemplo, poderia optar por fazer uma implementação parcial, sem objetivar a certificação.

Sendo assim, o trabalho de Camargo (2010) buscou identificar, através de um *survey* realizado com profissionais de teste, as práticas essenciais de um processo de teste para uma empresa de qualquer porte, gerando um *subset* de práticas como resultado.

A seguir, veremos um pouco mais sobre o *subset* proposto por Camargo (2010).

## 2.5 **Subset TMMi**

Modelos de referência e frameworks, como o TMMi, apontam o que pode ser feito para que o processo de teste possa ser melhorado, mas não indicam como fazê-lo. Apesar de ser organizado em níveis como o CMMI, e ser implantado de maneira incremental começando pelo nível mais baixo, o TMMi não indica quais áreas são prioritárias e podem ser implantadas primeiro, ou mesmo quais áreas podem deixar de ser implantadas caso o objetivo não seja obter a certificação.

Esta subseção relata um estudo que foi realizado por Camargo (2010) com o propósito de identificar quais são as práticas mais importantes do TMMi e que devem ser realizadas prioritariamente na execução de um processo de teste, segundo o ponto de vista de pessoas que trabalham há pelo menos três anos com teste de software em empresas que possuem um processo formal implantado. Acredita-se que existam atividades básicas relacionadas a cada fase de um processo de teste que não devem ser deixadas de lado, mesmo que o orçamento, tempo ou equipe sejam escassos.

Um *survey* é uma estratégia de investigação que permite a coleta de informações quantitativas e qualitativas por meio de questionários respondidos por uma amostra representativa da população que se deseja investigar. Os resultados do *survey* devem ser analisados para obter conclusões descritivas e explanatórias (WOHLIN, 2000).

O questionário foi dividido em seis grupos de questões. O primeiro grupo, relacionado ao perfil dos participantes, teve como objetivo identificar a experiência e o conhecimento, dos participantes, em modelos de referência, sendo eles o CMMI, o MPS.Br e o TMMi. Cada um dos cinco grupos restantes representam uma das fases do processo genérico de teste definido por Hohn (2011). São eles:

- A. Planejamento
- B. Projeto de Casos de Teste
- C. Configuração de Dados e de ambiente
- D. Execução e Análise
- E. Monitoramento e Controle

Cada página do questionário apresentava um único grupo de questões. Todas as questões relacionadas ao TMMi eram obrigatórias, de forma que o participante não poderia se abster de nenhuma resposta relacionada ao processo teste. Os participantes não foram informados a respeito da construção do questionário ter sido feita com base no TMMi.

Cada pergunta do questionário era um objetivo específico, que no TMMi possui práticas específicas relacionadas. Essas práticas eram as opções de cada pergunta, nas quais o respondente poderia atribuir uma nota, de 1 a 4 conforme a escala abaixo, de acordo com sua opinião pessoal em relação à importância daquela opção (prática) para a satisfação da pergunta descrita (objetivo específico).

- 1- Dispensável: Atividade dispensável que não precisa ser realizada.
- 2- Opcional: Atividade que não necessariamente precisa ser realizada.
- 3- Desejável: Atividade que deve ser realizada sempre que possível.
- 4- Mandatória: Atividade essencial que deve sempre ser realizada.

Com base na resposta desses participantes e nas análises feitas pela autora, pode-se observar que o conjunto de 81 práticas do TMMi pode ser reduzido a 33 (trinta e três) práticas, ainda assim mantendo a completude e coerência necessárias para a execução de um processo de teste. Isso representa uma redução em torno de 60% do TMMi. As outras práticas do TMMi não são descartáveis, porém, quando se tem como objetivo implementar um processo reduzido, ou ainda, quando a empresa não tem o *know-how* necessário para implementar sozinha seu próprio processo, ela pode utilizar esse conjunto de práticas para fazê-lo. Dessa forma, o subset representa uma maneira simplificada de criação ou melhoria de processos de teste, elaborada com base em um modelo de referência reconhecido e maduro.

O *subset* com as 33 práticas consideradas essenciais para um processo de teste é exibido a seguir, por fases:

- Planejamento:
  - Identificar riscos do produto;
  - Analisar riscos do produto;
  - Identificar elementos e características a serem testados;
  - Definir a abordagem de teste;
  - Definir critérios de parada;
  - Estabelecer cronograma de teste;
  - Estabelecer o plano de teste;
  - Obter necessidades do ambiente de testes;
  - Analisar requisitos do ambiente de testes;
- Projeto:
  - Identificar e priorizar casos de teste;
  - Identificar dados de teste específicos necessários;
- Configuração de dados e ambiente:
  - Desenvolver e priorizar procedimentos do teste;
  - Desenvolver o cronograma de execução do teste;
  - Implementar o ambiente de teste;
  - Realizar o teste intake;
- Execução e avaliação:
  - Executar casos de teste;
  - Relatar incidentes de teste;
  - Escrever log de teste;

- Decidir sobre incidentes com o grupo de controle de configuração;
  - Executar ações apropriadas para corrigir os incidentes de teste;
  - Acompanhar o status dos incidentes de teste;
  - Executar casos de teste não funcional;
  - Relatar incidentes de teste;
  - Escrever log de teste não funcional;
- Monitoramento e controle:
    - Conduzir revisões do progresso do teste;
    - Monitorar defeitos;
    - Conduzir revisões de qualidade do produto;
    - Analisar problemas;
    - Tomar ação corretiva;
    - Gerenciar ação corretiva;
    - Coordenar a disponibilidade e o uso dos ambientes de teste;
    - Relatar e gerenciar incidentes do ambiente de teste;

Este *subset* foi escolhido por representar, através do *survey* realizado por Camargo (2010), um modelo de processo enxuto, construído com base nas práticas do TMMi a partir da opinião de profissionais de teste de software. É importante ressaltar que o objetivo deste trabalho não é implantar um nível de maturidade do TMMi, portanto o *subset* é um modelo ideal para melhoria do processo de teste de uma organização de pequeno porte, como é o caso do NPI.

A seguir veremos os trabalhos que foram fundamentais para a construção desta monografia.

### 3 TRABALHOS RELACIONADOS

Camargo (2013), como citado anteriormente, identifica um *subset* de práticas do TMMi para estabelecer um processo de teste de software simplificado para pequenas e médias empresas. Esse processo simplificado baseia-se no resultado de um *survey*, realizado com profissionais de teste de software, que selecionou as práticas essenciais para um bom processo de teste em qualquer organização.

O trabalho citado assemelha-se a este através da definição da estratégia de elaboração de um processo simplificado, definindo um *subset* de práticas de acordo com as necessidades e características pertinentes à organização. Difere-se, entretanto, pelo meio em que esta estratégia será elaborada, uma vez que não será realizado um *survey* para identificar as



técnicas essenciais presentes no modelo TMMi. A elaboração da estratégia ocorrerá através de uma análise das atividades após a observação da organização e diagnóstico do processo de testes.

Soares (2013) apresenta a definição e implantação de um processo de software no Núcleo de Práticas de Informática (NPI) da Universidade Federal do Ceará. De acordo com a autora, o processo foi construído através da seleção de boas práticas de alguns modelos de processos de software tradicionais e ágeis, como CMMI, MPS.BR, RUP, PMBoK, Scrum e SCORE, e também através da análise das atividades desenvolvidas no NPI, tendo como foco as disciplinas de Gerência de Projetos, Requisitos e Gerência de Configuração.

Esse trabalho assemelha-se ao nosso através da definição e implantação de um processo de software, bem como o lugar onde o processo sugerido será implantado: o Núcleo de Práticas da UFC, Quixadá. Entretanto, esse trabalho trata de um processo completo, com várias disciplinas contempladas, enquanto o presente trabalho foca apenas no processo de testes.

Cruz (2010) por sua vez, realiza a avaliação do processo de teste de uma organização com base em um modelo de maturidade de teste de software. São analisados dois projetos, comparando suas práticas às práticas do processo de Gerência de Projetos de Teste de Software, do nível 1 do modelo MPT.BR. Após essa análise, verificou-se que a empresa possuía algumas boas práticas mas não aderiu ao nível do modelo analisado, então foram propostas algumas melhorias ao processo de teste.

Esse trabalho apresenta inicialmente o processo de escolha de um modelo de maturidade para que se baseiem as melhorias que serão propostas, sendo o MPT.Br o modelo escolhido posteriormente. Não é o caso deste trabalho, uma vez que o modelo que será utilizado como base é o de Camargo (2013) que provém do TMMi. Entretanto, a metodologia utilizada, bem como a avaliação do processo de teste na organização é semelhante a estratégia que será utilizada neste trabalho para que possamos cumprir nosso objetivo.

#### **4 PROCEDIMENTOS**

Esta seção apresenta os procedimentos metodológicos utilizados para este trabalho.

Primeiramente, é apresentada uma análise do ambiente onde o estudo de caso será realizado, no caso o Núcleo de Práticas em Informática da UFC. Posteriormente será estabelecido um processo de teste condizente com a realidade do NPI. Esta fase começa com a definição da estratégia para elaboração e aplicação de um processo de teste simplificado baseado no *subset* de práticas do TMMi. Em seguida, será feito o diagnóstico do processo de

teste atual da organização, para que se possa comparar o processo real ao modelo simplificado por Camargo (2013) e verificar se as práticas não atendidas pelo processo atual se adequam à realidade do NPI. Em seguida, se darão as fases de aplicação e implementação da proposta de processo de testes. Essas atividades estão melhor detalhadas a seguir.

#### **4.1 AMBIENTE EM ESTUDO**

A seguir será apresentada uma breve descrição do ambiente onde o estudo de caso foi realizado.

##### **4.1.1 Perfil do NPI**

O Núcleo de Práticas em Informática – NPI, da Universidade Federal do Ceará, campus de Quixadá, é uma organização de pequeno porte que fora criada em 2009 “com o objetivo de suprir com as necessidades de sistemas internos do campus” (GONÇALVES et al., 2013). O NPI foi escolhido por representar, de acordo com sua estrutura organizacional e finalidade, uma organização de pequeno porte. Em geral, empresas desse porte encontram maior dificuldade na elaboração de um processo de teste, principalmente pela falta de tempo e pela baixa importância dada a esse tipo de atividade (RODRIGUES; BESSA; PINHEIRO, 2010).

##### **4.1.2 Estrutura organizacional**

De acordo com Gonçalves et al. (2013) a equipe do NPI é distribuída da seguinte maneira:

- Professor supervisor realiza o acompanhamento das atividades supervisionando as atividades do gerente de projeto, dos líderes de projeto e das Equipes de projeto. Ele avalia os participantes de acordo com as entregas realizadas e desempenho individual.
- Gerente de projeto é responsável pela atividade de Gerenciamento do projeto, portanto deve realizar o planejamento e o acompanhamento das atividades a serem realizadas, além de remover os impedimentos.
- Líder de projeto é encarregado de definir requisitos e desenvolver funcionalidades do sistema, juntamente com a equipe do projeto. Além disto, o líder é responsável por tirar dúvidas dos membros da equipe do projeto relacionadas às questões tecnológicas e de garantir o envolvimento de todos.
- Equipe do projeto é responsável pelo planejamento das sprints, desenvolvimento e testes.

No entanto, como analista de requisitos e testes na empresa, foi possível perceber que os papéis e atividades são distribuídos de maneira um pouco diferente atualmente.

O NPI é uma fábrica de software desenvolvedora de sistemas de informação para fins internos, buscando suprir as necessidades do campus, contruindo em geral sistemas web. Possui uma equipe total de 32 pessoas, entre desenvolvedores, gerentes de projeto, líderes técnicos, analistas de requisitos e analistas de testes. O analista de teste, quando alocado para um projeto, é responsável por planejar e executar os testes.

O NPI funciona atualmente durante os períodos matutino e vespertino. Durante a tarde, o principal projeto é o projeto GPA-MAE (Gestão de Programas Acadêmicos – Módulo de Assuntos Estudantis). Durante a manhã possui três equipes alocadas para três projetos diferentes, sendo estes o GPA (Gestão de Programas Acadêmicos – Módulo de Projetos de Pesquisa), o projeto GAL (Gestão de Aquisição de Livros) e recentemente criado, o projeto GP (Gestão de Pessoas).

A seguir serão apresentados os passos para estabelecer a proposta de processo simplificado a ser aplicada.

## **4.2 ESTABELECIMENTO DO PROCESSO A SER APLICADO**

Nas subseções a seguir são descritas as etapas para que se possa estabelecer uma proposta de processo de teste para o NPI.

### **4.2.1 Definição de estratégia**

A estratégia para definição da proposta leva em consideração o contexto da empresa, utilizando como ponto de partida o conhecimento e a prática existentes na mesma. Isso permite definir um processo de teste que leva em conta as atividades já realizadas, objetivando melhorias sem causar grandes impactos na rotina de testes. O modelo genérico de processo de teste utilizado na etapa de diagnóstico – a seguir – indica quais práticas do TMMi são prioritárias para implantação. Portanto, indica para a empresa o que implantar primeiro para que se tenha um processo definido que pode ser controlado, avaliado e melhorado.

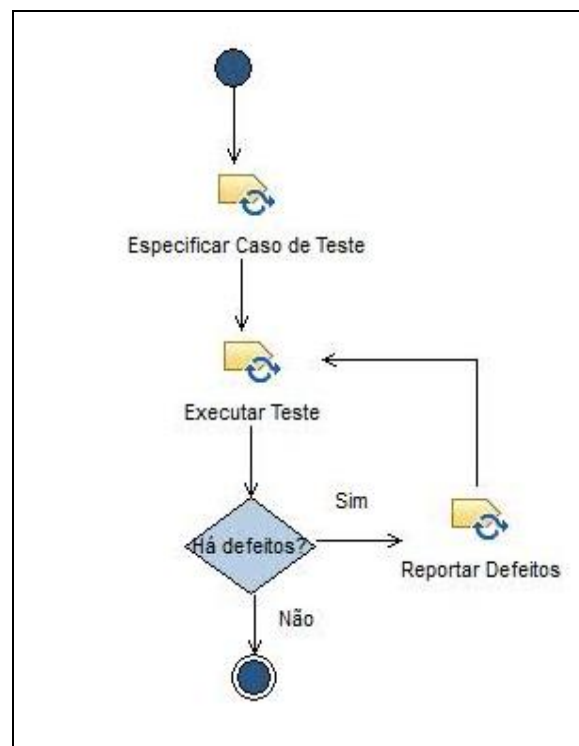
### **4.2.2 Diagnóstico do processo de testes atual**

Para conhecer melhor o processo de teste que o NPI possui, um diagnóstico foi realizado tendo como base de comparação o modelo genérico criado por Camargo (2013), com o intuito de identificar quais das práticas essenciais definidas por ela estão contempladas

no processo atual da empresa e como são realizadas, e obter um direcionamento inicial de quais delas devem ser implementadas. Também são identificados os papéis dos envolvidos e o tamanho da equipe de teste.

Para realizar essa etapa do trabalho, foi estabelecido como ponto de partida a modelagem do processo de teste desenhada pela própria organização, através da ferramenta *Eclipse Process Framework Composer* (EPF Composer), conforme a Figura 2.

Figura – 2 Processo de teste formal do NPI-UFC.



Fonte: NPI (201\_)

Cada atividade do processo possui um ou mais passos a serem realizados para que a atividade possa ser considerada concluída. Todas as atividades geram novos atributos em um único artefato, a Planilha de Teste, disponível no anexo B deste trabalho.

A primeira atividade do processo, “Especificar Caso de Teste”, tem o passo “Criar caso de teste”, que consiste em escrever os CT’s baseando-se no documento de especificação de casos de uso e no Backlog do Produto, inserindo-os na planilha de teste.

Em seguida, a atividade “Executar Teste” conta com o único passo “Executar testes”, que se dá pela execução dos casos de testes de acordo com os cenários descritos no documento de especificação de caso de uso, inserindo os resultados na planilha de teste. O passo indica que cada execução dos testes deve gerar uma nova aba na planilha.

Em caso de *bugs*, como demonstra o fluxo, aciona-se uma espécie de gatilho para a atividade “Reportar Defeitos”, que possui os seguintes passos:

- “Reportar defeitos”: onde os defeitos ocorridos durante a execução dos testes devem ser inseridos na planilha de testes. Ainda de acordo com o processo, a planilha de testes, que contém os problemas a serem corrigidos, deve ser adicionada ao Redmine (ferramenta de gerenciamento de projetos) 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.
- “Ações corretivas”: sugere que ações corretivas devem ser tomadas em relação aos defeitos reportados.

Por fim, o fluxo é finalizado.

### 4.2.3 Processo real

Durante o diagnóstico do processo de teste da empresa, foi possível identificar que algumas atividades que notadamente eram realizadas não estavam representadas no processo oficial desenhado pela empresa. O processo real contém 10 atividades, incluindo as atividades não documentadas no processo do NPI. Estas atividades estão listadas a seguir:

1. Definir a abordagem de teste;
2. Identificar elementos a serem testados;
3. Planejar a equipe de testes;
4. Criar casos de testes;
5. Implementar ambiente de teste;
6. Executar casos de teste;
7. Reportar defeitos;
8. Tomar ações corretivas;
9. Conduzir revisões de qualidade do produto;
10. Analisar problemas;

Essas atividades do processo real foram validadas pelos analistas de teste do NPI, tornando possível a realização da análise da aderência do processo explanada na subseção a seguir.

### 4.2.4 Análise da aderência do processo real ao *subset* do TMMi

A tabela indica quais práticas do *subset* estão contempladas no processo real da empresa, ou seja, o fluxo que ocorre de fato, mesmo que não documentado.

Tabela – 1 Aderência do processo real ao *subset*

<b>Atividades do processo real</b>	<b>Passos das atividades</b>	<b>Práticas aderentes ao <i>subset</i></b>	<b>Papel responsável</b>
<i>Não documentado</i>		1. Definir a abordagem de teste 2. Identificar elementos a serem testados 3. Planejar a equipe de testes	Gerente de Projeto
Especificar Caso de Teste	<ul style="list-style-type: none"> <li>• Criar caso de teste</li> </ul>	4. Identificar e priorizar casos de testes	Analista de Teste
<i>Não documentado</i>		5. Implementar ambiente de teste	Analista de Teste
Executar Caso de Teste	<ul style="list-style-type: none"> <li>• Executar casos de teste</li> </ul>	6. Executar casos de teste	Analista de Teste
Reportar Bugs	<ul style="list-style-type: none"> <li>• Reportar defeitos</li> <li>• Tomar ações corretivas</li> </ul>	7. Relatar incidentes de teste 8. Executar ações apropriadas para corrigir incidentes de testes	Analista de Teste
<i>Não documentado</i>		9. Conduzir revisões de qualidade do produto;	Gerente de Projeto
<i>Não documentado</i>		10. Analisar problemas;	Gerente de Projeto

Fonte: elaborada pela autora.

Na tabela, se pode observar que há dois perfis identificados que executam as atividades de teste: O Gerente de Projeto, que define alguns itens de planejamento, acompanha a atividade de teste e supervisiona o analista de teste, e o Analista de Teste, responsável pela implementação do ambiente de teste e pelo planejamento e execução dos testes.

Após uma verificação, viu-se que o processo real da empresa já contempla 10 das práticas consideradas essenciais pelo *subset*. Sendo pelo menos 4 destas não registradas no processo atual modelado pela empresa. De maneira geral, a empresa executa o fluxo modelado, porém não apresenta explicitamente todas as atividades realizadas pelo analista de teste e não contempla as atividades realizadas pelo gerente de projeto.

As seguintes práticas essenciais de acordo com o *survey* realizado por Camargo (2010) que não estão contempladas no processo real do NPI, por fase, são:

- Planejamento:
  - Identificar riscos do produto;
  - Analisar riscos do produto;

- Definir critérios de parada;
- Estabelecer cronograma de teste;
- Estabelecer o plano de teste;
- Obter necessidades do ambiente de testes;
- Analisar requisitos do ambiente de testes;
- Projeto:
  - Identificar dados de teste específicos necessários;
- Configuração de dados e ambiente:
  - Desenvolver e priorizar procedimentos do teste;
  - Desenvolver o cronograma de execução do teste;
  - Realizar o teste intake;
- Execução e avaliação:
  - Escrever log de teste;
  - Decidir sobre incidentes com o grupo de controle de configuração;
  - Executar ações apropriadas para corrigir os incidentes de teste;
  - Acompanhar o status dos incidentes de teste;
  - Executar casos de teste não funcional;
  - Relatar incidentes de teste;
  - Escrever log de teste não funcional;
- Monitoramento e controle:
  - Conduzir revisões do progresso do teste;
  - Monitorar defeitos;
  - Gerenciar ação corretiva;
  - Coordenar a disponibilidade e o uso dos ambientes de teste;
  - Relatar e gerenciar incidentes do ambiente de teste;

Após essa verificação, será analisada a viabilidade de aplicação dessas atividades no NPI junto ao Gerente de Projeto.

#### **4.2.5 Aplicação do modelo genérico**

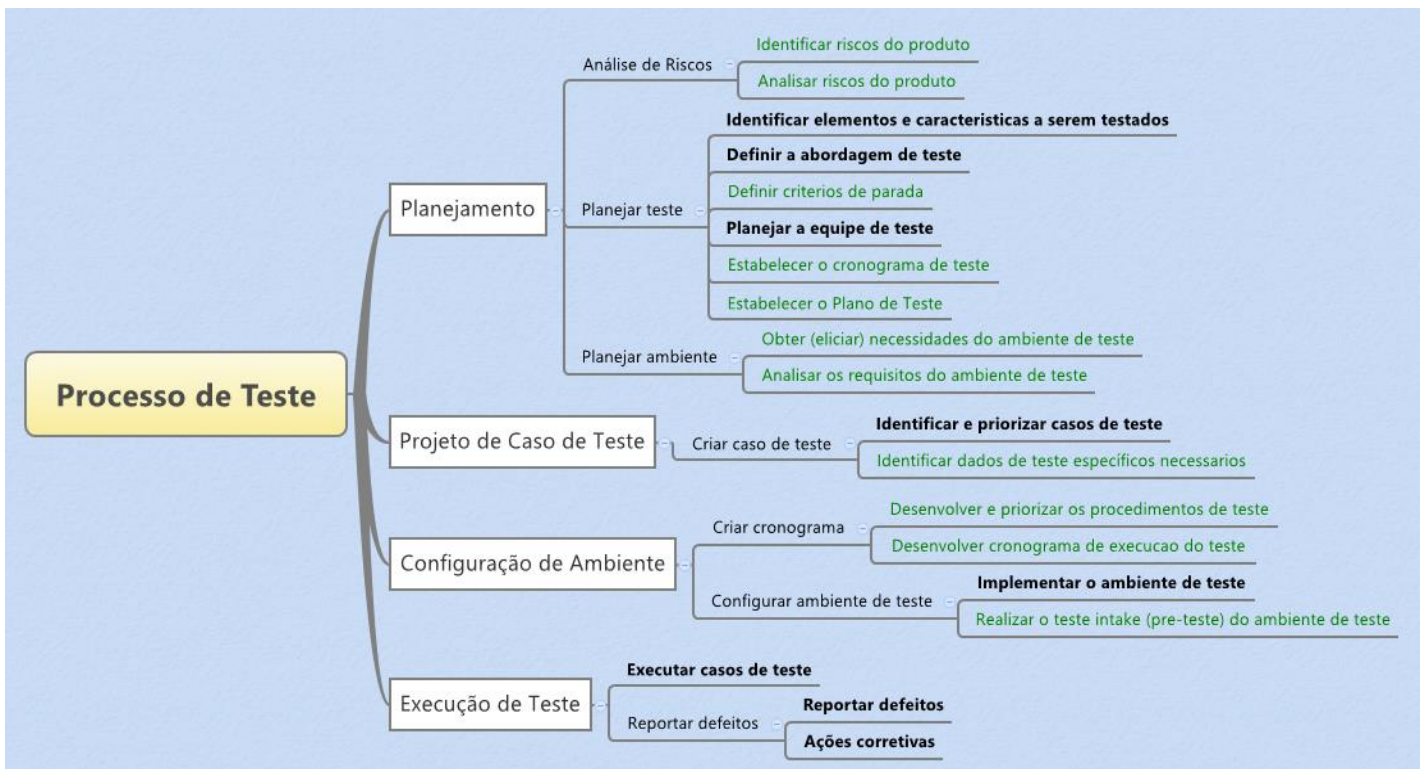
A aplicação do processo se dá pela disponibilização do processo de maneira complementar, ou seja, o processo padrão será reutilizado com adição de novas atividades. Foram definidas duas etapas para essa parte do trabalho, conforme veremos a seguir.

**Etapa 1** – Após a caracterização do processo atual da empresa, foi possível planejar a proposta de processo, recomendando a realização de atividades que foram consideradas pertinentes para a melhoria do processo de teste do NPI, de acordo com o *subset*.

A proposta de processo foi entregue e analisada pelo Gerente de Projeto, que sugeriu alguns ajustes para adequar a proposta à realidade do Núcleo de Práticas. E foi decidido junto à gerência que somente as práticas das fases de “Planejamento”, “Projeto” e “Configuração de ambiente” deveriam ser implementadas primeiro no projeto piloto.

**Etapa 2** – O modelo do processo de testes do NPI foi redesenhado incluindo as novas atividades que foram planejadas na etapa 1. Foram mantidas as atividades que já estão no processo real da empresa que correspondem a algumas das práticas do TMMi (em negrito na figura) e inseridas as atividades relativas às fases de Planejamento, Projeto e Configuração de ambiente (em verde na figura), gerando desta forma um processo mínimo que está de acordo com a realidade da empresa e causa baixo impacto na cultura da mesma, uma vez que não serão necessárias mudanças na equipe e na hierarquia atuais.

Figura – 3 Mapa mental da proposta de processo de teste para o NPI.



Fonte: elaborado pela autora.

O processo sugerido contém quatro fases: Planejamento, Projeto, Configuração e Execução, cada uma delas com suas respectivas atividades e passos. Considera-se que este processo contém as atividades e tarefas necessárias para que se satisfaça o conjunto mínimo de práticas do TMMi até a fase de Configuração de ambiente.



Papéis necessários para realização das atividades: Gerente de Projeto e Analista de Teste. Sobre a proposta, veremos o detalhamento das fases e tarefas a seguir:

**Planejamento:** Segundo a TMMi Foundation (2010), o proposta do planejamento de teste é definir uma abordagem de teste com base nos riscos identificados e definir uma estratégia de teste para estabelecer e manter um plano bem fundamentado para a realização e gerenciamento das atividades de teste. Na fase de planejamento também se identificam os recursos necessários, incluindo treinamentos, produtos a serem entregues, cronograma das atividades e o ambiente necessário para a realização dos testes (HASS, 2008).

Atividade/tarefa: Análise de Riscos

- Passos:
  - Identificar riscos do produto;
  - Analisar riscos do produto.
- Papéis envolvidos:
  - Gerente de Projeto.

De acordo com o survey realizado por Camargo (2013) a comunidade de testes concorda que a análise de risco do produto é importante para o sucesso do projeto de teste, pois é por meio dessa análise que se podem definir os casos de teste e priorizá-los.

Atividade/tarefa: Planejar Teste

- Passos:
  - Identificar elementos e características a serem testados;
  - Definir a abordagem de teste;
  - Definir critérios de parada;
  - Estabelecer o cronograma de teste;
  - Planejar a equipe de teste;
  - Estabelecer o Plano de Teste.
- Papéis envolvidos:
  - Gerente de Projeto;
  - Analista de Teste.

As práticas Identificar elementos e características a serem testadas, Estabelecer o cronograma de teste e Planejar a equipe de teste estão diretamente relacionadas com a prática Estabelecer o Plano de Teste, pois alguns dos itens que devem compor um plano de teste, segundo a norma 829 do IEEE (IEEE, 2008), são: o que será testado e o que não será testado;

a abordagem a ser utilizada; o cronograma; as classes e condições de teste identificadas; o critério de parada; entre outros.

Atividade/tarefa: Planejar Ambiente

- Passos:
  - Obter (eliciar) necessidades do ambiente de teste;
  - Analisar os requisitos do ambiente de teste.
- Papéis envolvidos:
  - Analista de Teste.

Na fase de Planejamento existem ainda práticas que estão relacionadas à definição do ambiente que será utilizado durante a execução do teste. São elas: Obter necessidades do ambiente de teste e Analisar requisitos do ambiente de teste. Essas duas práticas também estão relacionadas entre si, já que primeiro é necessário levantar os requisitos e necessidades específicas de ambiente do software que está sendo desenvolvido, para então analisar os requisitos levantados e sua viabilidade de implantação.

**Projeto:** Projeto de casos de teste é a fase seguinte ao planejamento e tem como entrada o plano de teste, que possui algumas definições essenciais para essa etapa como: análise de risco, o que será testado e abordagem de teste.

Atividade/tarefa: Criar Caso de Teste

- Passos:
  - Identificar e priorizar casos de teste;
  - Identificar dados de teste específicos necessários.
- Papéis envolvidos:
  - Analista de Teste.

**Configuração de Ambiente:** Como foi mostrado anteriormente na fase de planejamento, as necessidades de ambiente são identificadas e descritas para que possam ser implementadas. A fase de configuração de dados e de ambiente de teste possui como objetivo a implementação do teste e do ambiente de teste.

Atividade/tarefa: Criar Cronograma

- Passos:
  - Desenvolver e priorizar os procedimentos de teste;
  - Desenvolver cronograma de execução do teste.
- Papéis envolvidos:
  - Analista de Teste.

Segundo o TMMi, a prática Desenvolver e priorizar procedimentos de teste consiste em determinar a ordem de execução dos casos de teste de acordo com os riscos de produto que foram identificados. Esta priorização pode acontecer de acordo com agrupamentos, onde se pode priorizar os CT's por custo em caso de falha da funcionalidade, áreas de grande visibilidade (número grande de usuários afetados), prioridades para o cliente e variáveis como inclinação à falhas. A prática Desenvolver cronograma de execução do teste está diretamente relacionada à prática anterior, já que o cronograma depende da ordem de execução dos casos de teste.

Atividade/tarefa: Configurar Ambiente de Teste

- Passos:

Implementar o ambiente de teste;

Realizar o teste intake (pré-teste) do ambiente de teste.

- Papéis envolvidos:

Analista de Teste.

A prática Implementar o ambiente requer que se coloque em prática os requisitos de ambiente que foram analisados na fase de planejamento, enquanto Realizar o teste intake (pré-teste) do ambiente de teste significa certificar-se de que o ambiente está funcional.

**Execução:** Após a criação dos dados de teste, e do ambiente ter sido configurado para ser utilizado pela equipe de teste, o próximo passo é realizar a execução dos testes planejados. A equipe que foi alocada executa os casos de teste.

Atividade/tarefa: Executar Casos de Teste

Os casos de testes são executados pelo Analista de Testes de acordo com os cenários descritos no documento de especificação de caso de uso.

Atividade/tarefa: Reportar Defeitos

- Passos:

Reportar defeitos;

Ações corretivas.

- Papéis envolvidos:

Analista de Teste.

Na prática Reportar defeitos, após programar as correções, o gerente/líder técnico passa a monitorar os defeitos verificando o status do defeito, à medida que o analista realiza testes de regressão nos itens corrigidos. A prática Ações corretivas sugere que ações corretivas devem ser tomadas em relação aos defeitos reportados.

A próxima seção descreve a implementação da proposta de processo.

### 4.3 Implementação do modelo genérico

O processo de implementação tem como objetivo colocar o novo processo em funcionamento e em uso. A implementação do modelo genérico definido anteriormente se deu em duas etapas:

**Etapa 1** – Foi definido o projeto piloto para aplicação do novo processo de teste.

De início, foram disponibilizados todos os projetos do turno matutino para escolha de projeto-piloto para implementação da proposta. Foi escolhido pelo gerente, como projeto piloto, o projeto GPA (Gestão de Programas Acadêmicos). Outro projeto também foi colocado à disposição pela gerência, o GAL (Gestão de Aquisição de Livros).

**Etapa 2** – O novo modelo de processo de teste foi apresentado à equipe, juntamente à gerência. Posteriormente a proposta foi exposta em uma reunião separada com o analista de testes do projeto GAL, para que este pudesse entender e executar o processo de maneira adequada.

## 5 DESENVOLVIMENTO

### 5.1 Projeto-piloto GPA (Gestão de Programas Acadêmicos)

A primeira modificação oficial no processo se deu pela alteração do PGC (Plano de Gerência de Configuração), através da criação de um novo tipo de identificação da nomenclatura dos documentos no PGC. Atualmente não há nenhuma nomenclatura que especifique plano de testes e este não pode ser identificado por UC's (*use-cases*), como é o caso das especificações de requisitos.

Foi acordado com o gerente que o plano de teste seria identificado por [nome\_do\_projeto]-PLAN-TST. Este forneceu permissão de *push* no repositório da Wiki do NPI para a modificação.

Durante a execução do processo sugerido, como analista de teste do projeto, algumas questões puderam ser observadas. Mesmo as atividades que já eram realizadas no processo formal da empresa sofreram algumas mudanças.

Na fase de “Planejamento”, a atividade “Análise de Riscos” já era realizada de maneira intuitiva e não documentada, logo antes da criação de casos de teste, uma vez que os CT's eram criados de acordo com a capacidade de julgamento dos riscos pelo analista.

Durante a execução do processo proposto esta atividade continua sendo não documentada, mas foi possível garantir que o tempo e os recursos limitados foram usados para testar as áreas mais importantes do sistema, de maneira efetiva, já que esta fase está oficialmente no processo de teste e não depende mais da percepção do analista.

A atividade “Planejar Teste” também possui algumas tarefas que já eram realizadas, como se viu na figura 2. A mudança mais efetiva no processo se dá pela atividade “Estabelecer plano de teste”, onde são documentadas as decisões do gerente de projeto sobre o que deve ser testado, que abordagens utilizar e que membros serão alocados para esta atividade. Esta atividade não influi somente na criação de um novo artefato, mas em uma nova percepção de todo o processo realizado.

A atividade “Planejar Ambiente” não era realizada anteriormente e não raro causou transtornos para a equipe de teste. Como não havia nenhum planejamento para o ambiente de teste, o analista deveria fazer o que julgasse necessário para execução dos CT’s, assim, quando havia alguma restrição para execução do teste esta não era formalizada e geralmente a execução se dava de maneira errônea. Um exemplo disso é o uso do *browser* durante os testes funcionais, que por padrão deveria ser o Mozilla Firefox e muitas vezes era utilizado outro *browser* da preferência do analista, como o Google Chrome. Isso gerava interpretações erradas do teste, uma vez que algumas funcionalidades funcionavam perfeitamente no Google Chrome e não no Firefox, sendo assim não eram reportadas como falha na planilha de testes. Esse erro só foi descoberto na *Sprint Review*, ao demonstrar o sistema para o gerente de projeto.

Na fase de “Projeto”, a atividade “Criar caso de teste” contou com a priorização dos testes através dos riscos levantados na fase de planejamento, onde se decidiu o que testar usando as prioridades do cliente como uma maneira de medir o impacto da não realização dos casos de testes de algumas funcionalidades. O passo “Identificar dados de teste específicos necessários” continuou sendo realizado de acordo com a capacidade de julgamento da analista, uma vez que geralmente esses dados ficam implícitos na descrição dos CT’s na planilha de teste do NPI.

Na fase de “Configuração de Ambiente”, a atividade “Criar cronograma” foi documentada formalmente no Plano de Teste, onde foi definida uma ordem para a execução dos CT’s baseada nos riscos planejados na primeira fase do processo. No projeto, algumas funcionalidades tiveram de ser testadas na mesma ordem do fluxo de negócio desenhado pelo analista de requisitos. A partir daí se pôde construir o cronograma de execução, onde o modelo pode ser visto no anexo A deste trabalho.

A atividade “Configurar ambiente de teste” também foi de suma importância para o processo, uma vez que após elicitadas as necessidades de ambiente, ficaram muito mais fáceis e organizadas as dependências dos testes. “Realizar o teste *in-take* do ambiente” era um passo necessário que era feito de maneira informal, agora planejado e documentado se pode ter um controle mais rigoroso do ambiente.

As atividades da fase de “Execução” não foram adaptadas pelo *subset*, como acordado com o gerente, mas são sugeridas algumas melhorias na seção 5.3, uma vez que a atividade “Reportar Defeitos” não é refletida na planilha de testes do processo. Essa atividade é realizada através do *Redmine* (ferramenta de gerência de projetos) que permite criar *issues* para correção de *bugs* e alocá-los aos membros da equipe.

A seguir veremos o relato de aplicação do ponto de vista de outro analista de teste do NPI, no período matutino, responsável pelos testes no projeto GAL.

## 5.2 Projeto GAL (Gestão de Aquisição de Livros)

O projeto GAL conta atualmente com um analista de testes. Não foi necessário treinamento para que se pudessem realizar as novas atividades sugeridas na proposta de processo, uma vez que as mudanças causaram baixo impacto na rotina de testes da empresa. O analista respondeu a um questionário sobre seu perfil profissional tendo como objetivo identificar a experiência e o conhecimento do mesmo em processo de teste, modelos de referência, sendo eles o CMMI, o MPS.Br e o TMMi. Essas informações são consideradas importantes pois podem contribuir na curva de aprendizagem do processo proposto por este trabalho.

De acordo com as respostas, o analista tem conhecimento acadêmico sobre teste de software e trabalha com testes há cerca de sete meses no NPI, seguindo o processo de teste real. Possui conhecimento em MPS.Br e CMMI, e já sabia da existência do TMMi mas nunca havia trabalhado com o modelo na prática.

Durante a última sprint, o analista usou o processo de teste sugerido por esse trabalho. Foi realizada então uma entrevista não-estruturada com o analista para que ele pudesse expor as dificuldades que teve com o processo sugerido e suas percepções gerais sobre a proposta de processo.

Durante a fase de “Planejamento”, ele descreveu a dificuldade que teve no passo “Análise de Riscos”, onde ele não entendeu bem como essa análise deveria ser feita, mas em

contrapartida afirmou que entende que esse passo realmente tem um propósito importante para o processo como um todo.

No passo seguinte, ao “Planejar Teste”, ele sentiu dificuldade apenas com o passo “Definir critérios de parada”, pois não havia tido qualquer contato com essa atividade anteriormente. Sua visão sobre o Plano de Teste é de que este representa bem mais do que um artefato no processo, pois altera toda a percepção sobre testes e a maneira de executá-los, tornando o processo rigoroso e fácil de ser controlado.

No atividade “Planejar Ambiente”, ele concorda com a premissa de que é indispensável planejar o ambiente de execução antes de começar a executar os testes, evitando imprevistos durante a execução e erros de interpretação por parte do analista.

Durante a fase de “Projeto”, a atividade “Criar caso de teste” já era conhecida e realizada, mas sofreu algumas mudanças em sua concepção, ao contar com a análise de riscos feita na fase de planejamento.

Durante a fase de “Configurar Ambiente de Teste”, não ficou claro para o analista que ele deveria criar além do cronograma do teste, o cronograma de execução dos CT’s e, por isso, não realizou este passo da atividade no projeto. Foi sugerido que o template fornecesse informações mais claras e específicas para o estabelecimento deste cronograma.

Na fase de “Execução”, as atividades “Executar Casos de Teste” e “Reportar Defeitos” foram realizadas sem problemas e de maneira mais rápida, uma vez que segundo ele, após todo o planejamento estabelecido, os casos de teste foram melhor elaborados e os imprevistos com o ambiente de execução foram sanados.

## 6 DISCUSSÃO

O perfil geral dos membros do NPI é formado por alunos inexperientes e em formação, que estão geralmente em seu primeiro contato com o ambiente profissional. Isso interfere na curva de aprendizagem do processo, uma vez que a falta de maturidade e experiência com processos de desenvolvimento, em geral, afetam a maneira como eles executam o processo como um todo.

Ainda assim, a proposta de processo de teste contribuiu bastante para a rotina de testes no NPI, não só ao identificar e aplicar as atividades de teste condizentes com a realidade do núcleo, como ao definir uma ordem de execução dessas atividades, algo que o modelo de melhoria e o próprio *subset* não indicam.

Outro ponto positivo da proposta é que o processo também se adequa às atividades que ainda não são realizadas no NPI, mas que podem vir a ser implantadas. O uso de ferramentas

automatizadas e a necessidade de treinamento dos analistas de teste, por exemplo, são atividades que podem agregar ao processo sem precisar alterá-lo, uma vez que estes itens já estão previstos no Plano de Teste durante a fase de planejamento.

Ainda que o processo tenha sido adaptado de um modelo de melhoria já simplificado (através do *subset*), ele está de acordo com um modelo de referência específico para a fase escolhida, o que ressalta o enfoque em qualidade. Foi possível notar, mesmo que apenas em uma única execução na *sprint*, que o processo já apresenta melhorias, não só deixando de apresentar erros comuns no processo real, mas evitando que estes erros possam voltar a ocorrer.

A seguir veremos algumas das dificuldades e lições aprendidas durante a aplicação da proposta de processo, bem como possíveis melhorias a serem realizadas.

## 6.1 Dificuldades

São listadas a seguir as dificuldades encontradas durante a execução do processo proposto:

- Resistência a mudanças por parte de membros da equipe, uma vez que o pouco conhecimento em processo de software juntamente com questões culturais fez com que a importância em se utilizar um processo de teste tenha sido colocada em dúvida;
- Adequamento da proposta de processo à política ágil da empresa, pois foi necessário ter um cuidado especial para não tornar burocrático um processo que se baseia em um subconjunto simplificado de atividades de teste.

## 6.2 Lições Aprendidas

- É interessante escolher um projeto real para ser o projeto-piloto, um que seja considerado importante pela empresa e, de preferência, com um cliente representativo que aumente o comprometimento da empresa com a iniciativa de melhoria;
- É importante considerar os artefatos de projeto anteriores de modo a adequá-los para novos projetos, de forma a diminuir o esforço necessário para realização de atividades do processo;
- A inexperiência de algum membro do projeto torna algumas tarefas mais difíceis de serem compreendidas, sendo necessário um pouco mais de tempo para este habituar-se às novas atividades;



- É importante apresentar um modelo preenchido de um novo artefato proposto, além do tradicional *template*, para melhor entendimento do artefato pelo analista de teste.

### 6.3 Possíveis melhorias

Durante a execução deste trabalho algumas melhorias foram sugeridas para o processo:

- Adicionar uma aba “*Status*” na planilha de teste, para que a cada execução este *status* possa ser atualizado pelo analista de teste. Isso não inutilizaria o reporte de *bugs* na ferramenta Redmine, apenas tornaria mais prático visualizar o estado de caso de teste por cada funcionalidade testada.
- Adicionar um tópico “Cronograma de Execução” na seção “Cronograma” do Plano de Teste, para facilitar o entendimento do analista sobre criar o cronograma de execução dos casos de teste.

## 7 CONSIDERAÇÕES FINAIS

Com o crescente uso do software no dia a dia das pessoas, a exigência por um produto confiável e de qualidade é cada vez maior. Assim, a necessidade de realizar atividades de garantia de qualidade de software, como a atividade de teste, é cada vez mais evidente para as empresas manterem sua competitividade no mercado. Devido às dificuldades de implantação de um processo de teste em empresas de pequeno porte, este trabalho visou estabelecer um processo de teste simplificado no Núcleo de Práticas de Informática da UFC – Campus Quixadá, incluindo as boas práticas de um modelo de qualidade de software reconhecido no mercado.

O TMMi foi utilizado como base para a construção dessa proposta, pois é um modelo de referência específico para teste de software e permite que o processo elaborado tenha maior qualidade. Como a empresa possuía um processo em uso, ele foi levado em consideração e o processo proposto consistiu em uma melhoria do processo atual para que as práticas consideradas essenciais e adequadas à realidade do NPI, fossem acrescentadas.

Este trabalho buscou relatar as dificuldades e melhorias da aplicação do processo sugerido em um projeto piloto do NPI. A proposta refletiu em uma melhoria do processo como um todo, preparando o NPI para controlar e melhorar cada vez mais seu processo de teste e, conseqüentemente, seus produtos. O resultado deste trabalho infere na sugestão de validação da proposta do processo de teste apresentado, para o comitê responsável pelo processo de desenvolvimento do NPI.

A seguir são apresentadas as contribuições e limitações deste trabalho, bem como os possíveis trabalhos futuros a serem realizados.

## 7.1 Contribuições

Algumas das contribuições deste trabalho:

- O processo sugerido define os passos e a ordem para a execução de cada uma das atividades. Isso auxilia bastante no entendimento de como realizá-las;
- Um único artefato, o Plano de Teste, supre a maioria das demandas da fase de “Planejamento” e é simples de ser preenchido. Este artefato ainda influencia positivamente a Planilha de Testes que já era utilizada, resultando em uma melhor definição de casos de teste;
- Os responsáveis por cada atividade estão bem definidos;
- O incentivo à utilização do TMMi – um modelo de referência que tem se tornado cada vez mais conhecido pela comunidade brasileira devido ao crescente interesse pela atividade de teste – como fonte de informação para construção do processo;
- Aplicação de um processo de teste coerente com a cultura e a realidade da empresa, facilitando o seu aprendizado e continuidade pelos membros da equipe;
- A elaboração e o relato da aplicação de um processo de teste simplificado em uma empresa de pequeno porte real.

## 7.2 Limitações

Algumas das limitações deste trabalho:

- O processo sugerido ainda não foi validado pela empresa (exceto pela adição do Plano de Teste, que já foi oficialmente inserido ao processo);
- O processo sugerido contempla só até a terceira fase do *subset* proposto por Camargo (2010), não contando com as fases de “Execução e Avaliação” e “Monitoramento e Controle”;
- O *survey*, que deu origem ao processo simplificado, não obteve um número amplo de participantes e pode não retratar um consenso da comunidade de profissionais de teste.

### 7.3 Trabalhos Futuros

Algumas atividades podem ser realizadas como forma de contribuir para a melhoria do trabalho desenvolvido. Propõe-se que as seguintes atividades sejam realizadas para dar continuidade ao trabalho:

- Implementar as atividades da fase de “Execução” e “Monitoramento e Controle” do *subset*;
- Expandir o modelo de processo reduzido, identificando melhor os requisitos de cada prática, os produtos de trabalho esperados e as entradas e saídas do processo;
- Realizar um melhor acompanhamento da execução do processo.

## REFERÊNCIAS

- BELL, S. Lean Business-IT Integration, Part One: Who Wants to Go Talk to IT About This One? *Lean Enterprise Institute*, 2011.
- BURNSTEIN, I. Practical software testing: a process-oriented approach. New York: Springer, 2003, 709 p.
- CAMARGO, K. G. Elaboração de um Processo de Teste com Base em um Estudo de Caso Real. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de São Carlos, São Carlos, 2012.
- CAMARGO, K. G.; FERRARI, F. C.; FABBRI, S. C. P. F. Identifying a *Subset* of TMMi Practices to Establish a Streamlined Software Testing Process. *27th Brazilian Symposium on. IEEE*, p. 137-146, 2013.
- CROSBY, P. B. Qualidade é investimento: a arte de garantir a qualidade. 6.ed. Rio de Janeiro: José Olympio, 1994.
- CRUZ, G. A. Avaliação De Maturidade Em Processo De Teste De Software: Uma Pesquisa-Ação. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Lavras, Minas Gerais, 2010.
- ENGHOLM, H. J. Engenharia de Software na prática. 1ª ed. São Paulo: Novatec, 2010, 440 p.
- FUGGETTA, A. Software Process: A Roadmap, *The Future of Software Engineering*, ACM Press, p. 25-33, 2000.
- GONCALVES, E. J. T.; BEZERRA, C. I. M.; ALMENDRA, C. C.; SAMPAIO, A. L.; VASCONCELOS, D. R.. Núcleo de Práticas em Informática: Contribuindo para a Formação em Sistemas de Informação Através do Desenvolvimento de Projetos de Software. *XXI Workshop sobre Educação em Computação*. SBC, Macéio/AL, v. 1, p. 601-610, 2013.
- HERBERT, J. S. Teste Cooperativo de Software. Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 1999.
- HOHN, E. N. KITest: *Um arcabouço de conhecimento e melhoria de processo de teste*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP - Brazil, jun. 2011
- HUMPHREY, W. S. Managing the software process. *Addison-Wesley Professional*, 1989.
- JINO, M.; MALDONADO, J. C.; DELAMARO, M. E. Introdução ao teste de software. São Carlos: Elsevier, 2007.
- KOSCIANSKI, A.; SOARES, M. S. Qualidade de Software. 2. ed. São Paulo: Novatec, 2007.
- PRESSMAN, R. S. Engenharia de Software. 6. ed. São Paulo: McGraw Hill, 2006, 711 p.

MALDONADO, J. C.; FABRI, S. C. Verificação e Validação de Software. *Qualidade de Software—Teoria e Prática*, São Paulo: Prentice Hall, p. 66-73, 2001.

MPS.BR. Melhoria de Processo do Software Brasileiro - Guia Geral. [S.l.], 2011.

MPT.BR. Melhoria do Processo de Teste Brasileiro. [S.l.], 2011.

MYERS, G. J. et al. The art of software testing. 2. ed. New Jersey: Wiley, 2004, 224 p.

RODRIGUES, A.; BESSA, A.; PINHEIRO, P. R. Barriers to implement test process in small-sized companies. *Organizational, Business, and Technological Aspects of the Knowledge Society*. Springer Berlin Heidelberg. p. 233-242, 2010.

SIQUEIRA, J. O Modelo de Maturidade de Processos: como maximizar o retorno dos investimentos em melhoria da qualidade e produtividade, *60º ABM CONGRESS*, 2005.

SEI. Capability Maturity Model Integration Version 1.2. Guia Geral: 2006.

SOARES, M. P. et al. Definição e Implantação de um Processo de Software para o Núcleo de Práticas de uma Universidade. Monografia (Graduação em Sistemas de Informação) – Universidade Federal do Ceará, Quixadá, 2013.

SOFTEX. MPS.BR - Melhoria de Processo do Software Brasileiro: Guia Geral: 2009.

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Education, 2011, 529 p  
TMMi Foundation. Portal Corporativo. 2014.

WEBER, K. et al. Melhoria de Processo do Software Brasileiro (MPS. BR): um programa mobilizador. *XXXI Conferencia Latinoamericana de Informatica*. CLEI, Santiago, 2006.

WEINBERG, G. M. Software com qualidade: pensando e idealizando sistemas. São Paulo: Makron Books, 1993, 387 p.

WOHLIN, C. *Experimentation in software engineering: an introduction*. [S.l.]: Springer, 2000. ISBN 9780792386827.

## ANEXOS

### ANEXO A – Plano de Teste

# Plano de Teste

## <Nome do Projeto>

### HISTÓRICO DE REVISÃO

Data	Versão	Responsável	Alteração

Fonte: adaptado do *Rational Unified Process* (RUP).

## 1. INTRODUÇÃO

### 1.1 Escopo

*Descreva os níveis de teste (por exemplo, Unidade, Integração ou Sistema) e os tipos de teste (como, por exemplo, Funcionalidade, Usabilidade e Desempenho) que serão abordados por este **Plano de Teste**. Também é importante fornecer uma indicação geral das áreas importantes que serão **excluídas** do escopo, especialmente nos casos em que o público-alvo possa supor sensatamente que elas serão incluídas.*

### 1.2 Objetivos

*Descreva a finalidade do **Plano de Teste**.*

### 1.3 Público Alvo deste Documento

*Forneça uma breve descrição do público para o qual o **Plano de Teste** está sendo escrito. Isso ajudará os leitores do documento a identificarem se ele realmente está destinado ao seu uso e também ajudará a evitar que o documento seja usado de forma inadequada.*

## 2 PROPÓSITO

### 2.1 Tipos e técnicas de teste

*Descreva que tipos de teste (por exemplo: funcional, não-funcional, regressão) e técnicas de teste (por exemplo: exploratório, ad hoc) serão utilizados.*

### 2.2 Itens e Funcionalidades

*Identifique quais itens e funcionalidades serão testados.*

## 3 ORGANIZAÇÃO

### 3.1 Recursos humanos e responsabilidades

*Esta seção apresenta os recursos necessários para abordar o esforço de teste e as principais responsabilidades.*

Papel	Quantidade Mínima	Responsabilidade

#### 3.1.1 Capacitação

*Identifique a necessidade de capacitação para os recursos humanos, relacionada a determinada tecnologia ou ferramenta.*

### 3.2 Necessidades De Ambiente

*Identifique os recursos não humanos necessários ao **Plano de Teste**.*

Nome do Elemento de Software	Versão	Tipo e Outras Observações

#### 3.2.1 Ferramentas

*Identifique a necessidade de uso de ferramentas para a execução dos testes.*

## 4 CRITÉRIOS DE ACEITE E DE PARADA

### 4.1 Critérios de aceite

*Especifique os critérios que serão usados para determinar se a execução do **Plano de Teste** foi concluída ou se a continuação da execução não será vantajosa.*

### 4.2 Critérios de parada

*Especifique os critérios que serão usados para determinar se os testes deverão ser prematuramente suspensos ou concluídos antes que o plano tenha sido totalmente executado. Especifique também segundo que critérios os testes poderão ser reiniciados.*

## 5 CRONOGRAMA

*Contém uma descrição de marcos importantes (milestones) das atividades (incluindo as datas de início e fim da atividade). Apenas marcos relevantes devem ser listados, ou seja, aqueles que contribuirão nas atividades de testes.*

<b>Milestone</b>	<b>Início</b>	<b>Término</b>	<b>Responsável</b>

## 6 PRODUTOS

### 6.1 Artefatos de entrada

*Esta seção apresenta a documentação necessária para a produção do **Plano de Teste**.*

### 6.2 Artefatos de saída

*Esta seção apresenta a documentação gerada a partir do **Plano de Teste**.*



## ANEXO B – Print-screen da Planilha de Teste do NPI.

Microsoft Excel - GPA-IST-ESPEC-UC01-cadastrar_projeto.xlsx									
1	A	B	C	D	E	F	G	H	I
2	Casos de Uso	Fluxo	Casos de Teste - Identificação	Cenário de Teste	Passos	Resultados Esperados	Dados de Entrada	Resultado da Execução	Observações
4	Cadastrar Projeto	CRIAR	TC_01_01	Usuário preenche todos os campos obrigatórios.	1. Usuário seleciona Novo Projeto na tela inicial. 2. O usuário preenche o campo Nome do Projeto com um valor válido. 3. O usuário preenche o campo Descrição do Projeto com um valor válido. 1. Usuário seleciona Novo Projeto na tela inicial. 2. O usuário não preenche o campo Nome do Projeto. 3. O usuário não preenche o campo Descrição do Projeto.	Projeto cadastrado no banco e mensagem de indicação de sucesso exibida na tela.	Nome: TC0101 Projeto OK Descrição: Descrição Teste OK	PASSOU PARCIALMENTE	Não exibiu mensagem de confirmação.
5	Cadastrar Projeto	CRIAR	TC_01_02	Usuário não preenche nenhum dos campos obrigatórios.	1. Usuário seleciona Novo Projeto na tela inicial. 2. O usuário preenche o campo Nome do Projeto. 3. O usuário não preenche o campo Descrição do Projeto.	Projeto não cadastrado no banco e mensagem de indicação para preenchimento dos campos obrigatórios exibida na tela.	Nome: (vazio) Descrição: (vazio)	PASSOU	Exibiu mensagem mínimo de caracté para o campo Nome do Projeto
6	Cadastrar Projeto	CRIAR	TC_01_03	Usuário cadastra um projeto com um nome já cadastrado no banco.	1. Usuário preenche o campo Nome do Projeto com um valor já existente no banco. 3. O usuário preenche o campo Descrição do Projeto com um valor válido.	Projeto cadastrado no banco e mensagem de indicação de sucesso exibida na tela.	Nome: TC0101 Projeto OK Descrição: Descrição Teste OK	PASSOU PARCIALMENTE	Não exibiu mensagem de confirmação.
7	Cadastrar Projeto	CRIAR	TC_01_04	Usuário preenche campo Nome do Projeto somente com caracteres inválidos.	1. Usuário seleciona Novo Projeto na tela inicial. 2. O usuário preenche o campo Nome do Projeto inserindo somente caracteres inválidos. 1. Usuário seleciona Novo Projeto na tela inicial.	Projeto não cadastrado no banco e mensagem de indicação para preenchimento dos campos obrigatórios.	Nome: (vazio) Descrição: Descrição Teste OK	PASSOU PARCIALMENTE	Não exibiu mensagem sobre o erro.
8	Cadastrar Projeto	CRIAR	TC_01_05	Usuário preenche campo Descrição do Projeto somente com caracteres inválidos.	1. Usuário seleciona Novo Projeto na tela inicial. 2. O usuário preenche o campo Descrição do Projeto inserindo somente caracteres inválidos.	Projeto não cadastrado no banco e mensagem de indicação para preenchimento dos campos obrigatórios.	Nome: TC0105 Projeto OK Descrição: (vazio)	FALHOU	Projeto cadastrado Não exibiu mensagem sobre o erro.
9	Cadastrar Projeto	CRIAR	TC_01_06	Usuário preenche campo Participantes com valor inválido.	1. Usuário seleciona Novo Projeto na tela inicial. 2. O usuário preenche o campo Participantes inserindo números.	Projeto não cadastrado no banco e mensagem de indicação de valor inválido.	Nome: TC0106 Projeto OK Descrição: Descrição Teste OK Participantes: 3	FALHOU	Projeto cadastrado Não exibiu mensagem sobre o erro.
10	Cadastrar Projeto	CRIAR	TC_01_07	Usuário preenche campo Participantes com caracteres inválidos.	1. Usuário seleciona Novo Projeto na tela inicial. 2. O usuário preenche o campo Participantes inserindo caracteres especiais.	Projeto não cadastrado no banco e mensagem de indicação de caracteres não permitidos.	Nome: TC0107 Projeto OK Descrição: Descrição Teste OK Participantes: *	FALHOU	Projeto cadastrado Não exibiu mensagem sobre o erro.
11	Cadastrar Projeto	CRIAR	TC_01_08	Usuário preenche campo Data com valor inválido.	1. O usuário preenche o campo Data de início com valor maior que Data de Término.	Projeto não cadastrado no banco e mensagem de indicação de data inválida.	Nome: TC0108 Projeto OK Teste início: 06/05/2014 Teste Término: 06/05/2013 Descrição: Descrição Teste OK	FALHOU	Projeto cadastrado Não exibiu mensagem sobre o erro.