

**GABRIELA DOS SANTOS SILVA OLIVEIRA**

**MELHORIA CONTÍNUA DE PROCESSOS EM MANUTENÇÃO DE SOFTWARE  
DO MERCADO DE INVESTIMENTOS**

Monografia apresentada ao PECE –  
Programa de Educação Continuada em  
Engenharia da Escola Politécnica da  
Universidade de São Paulo como parte dos  
requisitos para conclusão do curso de MBA  
em Tecnologia de Software.

São Paulo

2017

**GABRIELA DOS SANTOS SILVA OLIVEIRA**

**MELHORIA CONTÍNUA DE PROCESSOS EM MANUTENÇÃO DE SOFTWARE  
DO MERCADO DE INVESTIMENTOS**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientadora: Profa. Dra. Jussara Pimenta Matos

São Paulo  
2017

## **AGRADECIMENTOS**

Agradeço a Deus pela força e disposição. A minha família pela paciência com a minha ausência e ao apoio nesses anos de MBA. E a todos os professores que contribuíram com a minha formação acadêmica e pessoal.

## RESUMO

A proposta deste trabalho é aplicar melhoria nos processos da área de manutenção de software de uma empresa do mercado de investimentos. Este mercado está em constante mudança, logo os seus produtos precisam se adaptar continuamente. Devido a dinâmica existente nesse mercado, as soluções de problemas e inclusão de novas funções nos softwares podem comprometer os requisitos não funcionais. O acompanhamento de mudanças em softwares deve seguir um processo bem definido, adequado e flexível, o que é um grande desafio para áreas que estão em constante mudança. A aplicação de melhoria nos processos de manutenção de software é realizada, conforme o processo Gerência de Projetos do modelo Melhoria de Processo do Software Brasileiro (MPS.BR) nível G. Ao final são coletadas informações sobre as atividades realizadas na organização através da implementação do processo, e os resultados alcançados.

**Palavras-chave:** Manutenção de Software. Processo de Software. Melhoria de Processo de Software (SPI). Melhoria de Processo do Software Brasileiro MPS.BR.

## ABSTRACT

The proposal of this work is to apply improvement in the processes of the software maintenance area of an investment market company. This market is constantly changing, so your products need to adapt continuously. Due to the dynamics in this market, problem solving and inclusion of new functions in software may compromise non-functional requirements. Tracking software changes must follow a well-defined process, adequate and flexible, which is a great challenge for areas that are constantly changing. The application of improvement in software maintenance processes is performed, according to the Project Management Process of the Brazilian Software Process Improvement (MPS.BR) level G. At the end, information is collected on the performed activities in the organization through the implementation of the process, and the results achieved.

**Keyword:** Software Maintenance. Software Process. Software Process Improvement (SPI). Melhoria de Processo do Software Brasileiro MPS.BR.

## **LISTA DE ILUSTRAÇÕES**

	Pág.
Figura 1 – Custo de cada tipo de manutenção	10
Figura 2 – Níveis de maturidade do People CMM	24
Figura 3 – Estrutura de TI com foco em manutenção de software	33
Figura 4 – Fluxo de abertura de chamados	40
Figura 5 – Atividades de manutenção de software	43
Figura 6 – Papéis e responsabilidades na implementação MPS.BR	50
Figura 7 – Ciclo de manutenção de software na organização	52
Figura 8 – Papéis e responsabilidades na organização	55

## **LISTA DE TABELAS**

	Pág.
Tabela 1 – Modelo CMMI	13
Tabela 2 – Modelo MR-MPS-SW	15
Tabela 3 – Filosofia do People CMM	23
Tabela 4 – Modelo People CMM	25
Tabela 5 – Resultado dos projetos de software de 2011 a 2015	27
Tabela 6 – Método Ágil X Cascata de 2011 a 2015	28
Tabela 7 – Exemplo de ocorrência por categoria	37
Tabela 8 – Problemas na área de manutenção de software da organização	47
Tabela 9 – Cronograma de planejamento de atividades	48
Tabela 10 – Tempo de conclusão dos chamados	51

## **LISTA DE ABREVIATURAS E SIGLAS**

CMMI	Capability Maturity Model Integration
CMMI-SVC	Capability Maturity Model Integration for Services
ETL	Extract, Transform, Load
ITIL	Information Technology Infrastructure Library
MA-MPS	Método de Avaliação MPS
MPS.BR	Melhoria de Processo do Software Brasileiro
MPS-SV	Modelo do Processo de Serviços
People CMM	People Capability Maturity Model
SEI	Software Engineering Institute
SGBD	Sistema de Gerenciamento de Banco de Dados
SPI	Software Process Improvement
SQA	Software Quality Assurance
TI	Tecnologia da Informação

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>1</b>
1.1	MOTIVAÇÃO .....	2
1.2	OBJETIVO .....	3
1.3	JUSTIFICATIVAS.....	3
1.4	ESTRUTURA DO TRABALHO .....	3
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA.....</b>	<b>5</b>
2.1	CONSIDERAÇÕES INICIAIS .....	5
2.2	PROCESSO DE SOFTWARE .....	5
2.3	MANUTENÇÃO DE SOFTWARE .....	7
2.4	PROCESSO DE MANUTENÇÃO DE SOFTWARE .....	8
2.5	MELHORIA DE PROCESSOS DE SOFTWARE.....	12
2.5.1	<i>Modelos de processos de melhoria.....</i>	12
2.5.2	<i>Problemas na implantação de melhoria de processos .....</i>	21
2.5.3	<i>O papel das pessoas .....</i>	22
2.6	QUALIDADE DE SOFTWARE .....	26
2.7	FATORES DE SUCESSO PARA PROJETOS DE SOFTWARE .....	27
2.8	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	29
<b>3</b>	<b>DESENVOLVIMENTO .....</b>	<b>30</b>
3.1	CONSIDERAÇÕES INICIAIS .....	30
3.2	ÁREAS DE TECNOLOGIA DA INFORMAÇÃO DA ORGANIZAÇÃO.....	31
3.3	ÁREA DE MANUTENÇÃO DE SOFTWARE DA ORGANIZAÇÃO.....	34
3.3.1	<i>Equipe de manutenção de software.....</i>	34
3.4	PROCESSO DE MANUTENÇÃO DE SOFTWARE .....	35
A.	<i>Classificação e categorização.....</i>	36
B.	<i>Tempo previsto para a conclusão do chamado.....</i>	38
C.	<i>Estado do chamado.....</i>	38
D.	<i>Abertura de chamados.....</i>	39
E.	<i>Fluxo de melhorias .....</i>	40
3.5	ATIVIDADES DE MANUTENÇÃO DE SOFTWARE .....	41
3.6	PROBLEMAS NA ÁREA DE MANUTENÇÃO DE SOFTWARE .....	44
3.7	PLANEJAMENTO DE ATIVIDADES PARA A IMPLEMENTAÇÃO DO PROCESSO GERÊNCIA DE PROJETOS DO MPS.BR NÍVEL G NA ORGANIZAÇÃO .....	47

3.8	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	49
<b>4</b>	<b>ANÁLISE DOS RESULTADOS .....</b>	<b>50</b>
4.1	CONSIDERAÇÕES INICIAIS .....	50
4.2	IMPLEMENTAÇÃO E RESULTADOS DO PROCESSO GERÊNCIA DE PROJETOS .....	50
4.3	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	60
<b>5</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>61</b>
5.1	CONTRIBUIÇÕES DO TRABALHO .....	61
5.2	TRABALHOS FUTUROS .....	61
	<b>REFERÊNCIAS.....</b>	<b>63</b>

## 1 INTRODUÇÃO

O software tem conquistado um papel essencial e crítico em nossa sociedade. A dependência relacionada aos recursos de sistemas automatizados, está cada vez maior. Ao mesmo tempo as aplicações de software são produtos complexos, difíceis de desenvolver e testar, e por muitas vezes exibem comportamentos inesperados e indesejados, originando problemas.

A Engenharia de Software motiva a criação de processos, métodos e ferramentas, afim de construir produtos de qualidade, dentro do prazo e tempo previstos e reduzir os custos com a manutenção (Sommerville, 2011). Os resultados do *2015 Chaos Report*, estudo realizado em 50 mil projetos em todo o mundo, indicam que ainda há bastante trabalho a ser feito para alcançar resultados bem sucedidos em projetos de desenvolvimento de software. O estudo aponta que a maior parte dos softwares foi entregue com algum problema de requisitos, prazo ou custo (Lynch, 2015).

Os processos possuem o objetivo de determinar as atividades a serem executadas durante a construção ou modificação de um software. Além de fornecer estabilidade, controle e organização para atividades, que se não controladas, poderiam se tornar um caos (Pressman, 2010).

Normalmente, a maior parte dos gastos da área de Tecnologia da Informação (TI) nas organizações, é direcionada a manutenção de software (Zetlin, 2013), que se refere principalmente ao desenvolvimento contínuo, ou seja, a evolução do software como também a correção de erros de codificação, especificação ou de projeto (Sommerville, 2011). Infelizmente, a maioria das mudanças nos softwares vêm com alguma deterioração (Mens, 2008), por isso as filas para correções de *bugs* são crescentes, assim como, as solicitações de melhorias (Pressman, 2010). A norma IEEE 14764-2006 foi criada devido a necessidade de um processo para organização e gestão das atividades de manutenção de software. E possui métodos para classificar, priorizar, realizar e medir as atividades pertinentes a manutenção, facilitando na gestão das demandas e recursos (Croll; Pigoski; Moore, 2006).

Os principais objetivos do *Software Process Improvement* (SPI) são entender os processos atuais, implantar mudanças nos processos afim de melhorar sua qualidade, e reduzir os custos ou acelerar o tempo de realização das atividades

(Srivastava; Awasti, 2014). Existem modelos padrões de melhoria de processos, como por exemplo *Capability Maturity Model Integration* (CMMI) e Melhoria de Processo do Software Brasileiro (MPS-BR). Os modelos servem como uma base, sendo necessário adaptá-los ao contexto da organização, e dependem principalmente do esforço humano para sua efetiva aplicação (Kuhrmann; Diebold; Munch, 2016). Problemas comuns para a implantação de melhoria de processos, estão relacionadas principalmente a falta de apoio da gestão, dedicação insuficiente nas atividades e falta de experiência dos recursos envolvidos, em boas práticas de Engenharia de Software (Pressman, 1996).

O *People Capability Maturity Model (People CMMI)* é um modelo de processos direcionado aos recursos humanos. O modelo evidencia a importância da capacitação da força de trabalho, através das competências individuais de cada pessoa, para o sucesso no desempenho e realização de tarefas dos processos e melhoria dos processos (Matturo; Saavedra, 2012).

A implantação de boas práticas de Engenharia de software, quando bem sucedida, resulta em projetos mais propensos a serem concluídos a tempo, com melhor comunicação, o caos é reduzido, os erros encontrados pelo cliente diminuem e a credibilidade da organização aumenta (Pressman, 2010). Processos eficientes são inúteis, se não há pessoas preparadas para a realização das atividades, por isso é importante considerar as habilidades, práticas e características organizacionais necessárias para a execução dos processos (Haron *et al.*, 2013).

## 1.1 Motivação

A participação da autora da monografia em um projeto de melhoria de processos na área manutenção de software é a motivação principal para a escolha desse tema. A equipe passava por uma fase de caos, impactando no desempenho dos analistas e na gestão dos projetos. A partir de ações de melhoria contínua nos processos foi possível observar mudanças positivas nos resultados da área de manutenção de software da organização. Dessa forma, o trabalho foi direcionado para

a pesquisa e estudo de caso sobre melhoria em processos de manutenção de software no setor de atuação da empresa, isto é, a área de investimentos.

## 1.2 Objetivo

O objetivo deste trabalho é apresentar uma proposta de ações para melhoria contínua de processos em projetos de manutenção de software de uma empresa do mercado de investimentos, e sua aplicação. Para isso, são abordados os principais conceitos, como processos de software, manutenção e processos de manutenção. O conhecimento adquirido na etapa teórica é a base para a proposta de melhoria dos processos da empresa, onde o estudo de caso é realizado.

## 1.3 Justificativas

É grande a importância da utilização de software pelas organizações, pois este se tornou uma ferramenta fundamental de execução de tarefas e integração entre diferentes áreas. Grande parte das negociações têm como origem o desenvolvimento de um software.

Esse trabalho apresenta a importância do software e qual a relevância dos processos para o desenvolvimento e a manutenção do software, como eles são criados, qual o papel das pessoas relacionado aos processos e o que é processo de melhoria contínua.

## 1.4 Estrutura do Trabalho

Esta monografia é estruturada em cinco capítulos.

O capítulo 1 INTRODUÇÃO, apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

O capítulo 2 REVISÃO BIBLIOGRÁFICA apresenta os conceitos teóricos, abordando assuntos sobre os processos de desenvolvimento e manutenção de software, melhoria nos processos, o papel e responsabilidades das pessoas, o que é

qualidade de software, e quais fatores podem contribuir para o sucesso em projetos de software.

O capítulo 3 DESENVOLVIMENTO dedica-se ao desenvolvimento da pesquisa e estudo de caso. São descritos os serviços realizados pelo mercado de investimentos, as características da área de manutenção de software, os processos da organização e o papel das pessoas. E então é apresentando um planejamento para a melhoria contínua de processos para a área de manutenção de software para a organização estudada, conforme as características do processo Gerência de Projetos do MPS.BR nível G.

O capítulo 4 ANÁLISE DOS RESULTADOS apresenta os resultados obtidos com a aplicação do modelo de melhoria contínua em processos de manutenção de software ao estudo de caso.

O capítulo 5 CONSIDERAÇÕES FINAIS apresenta a conclusão do autor referente ao trabalho, as contribuições, e as propostas para trabalhos futuros baseados no que foi realizado.

REFERÊNCIAS relacionam as fontes de consulta para a elaboração dessa monografia.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Considerações iniciais

A Engenharia de Software surgiu com a intenção de propor boas práticas às áreas de tecnologia, e motiva a criação e evolução de processos, métodos e ferramentas, afim de construir software de qualidade, dentro do prazo e tempo previstos (Sommerville, 2011).

Apesar da evolução de conceitos propostos pela Engenharia de Software, o processo de desenvolvimento e manutenção de software enfrentam problemas até hoje (Lynch, 2015). A dinâmica do mercado e da tecnologia é acelerada, novos conceitos surgem, o cenário da organização é transformado, e as pessoas e os softwares precisam ser adaptados a essa dinâmica.

O mercado de investimentos, contexto em que está inserido o trabalho, está em constante transformação e depende do comprometimento e experiência das pessoas, para que as mudanças nos softwares possam aumentar o seu potencial competitivo, promovendo um trabalho de valor aos seus clientes.

Há alguns anos a autora dessa monografia atua na área de manutenção de software em uma empresa do setor de investimentos, onde acompanha a melhoria de processos. Importante fator de contribuição para a realização dessa monografia.

### 2.2 Processo de software

Processo de software é como uma estrutura, um roteiro para as atividades, ações e tarefas necessárias para a construção de um software de alta qualidade. Essas atividades podem envolver o desenvolvimento de software a partir do zero, ou a ampliação e modificação de um software existente (Pressman, 2010; Sommerville, 2011). O processo é um dos mais importantes fatores de sucesso para um projeto de software (Haron *et al.*, 2013). Através da utilização de processos é possível perceber que o desenvolvimento de software é algo complexo, e que a qualidade de um produto

de software depende das pessoas, da organização e dos procedimentos usados para criar e entregar o software (Fuggetta, 2000).

Modelos de processos são uma representação simplificada de um processo de software. Cada modelo representa um processo sob uma perspectiva em particular. Sommerville (2011) divide os processos em duas categorias: planejados ou processos ágeis. Para os processos orientados a um plano, as atividades são planejadas antecipadamente e o progresso é medido referente ao plano. Nos processos ágeis o planejamento é incremental, e é mais fácil mudar o processo para refletir as necessidades dos clientes. Ele considera quatro atividades como fundamentais para a Engenharia de Software, que de alguma forma fazem parte de todos os processos de software: especificação, projeto de software, implementação, validação e evolução. Ele destaca três modelos de processos:

- Modelo cascata: as atividades fundamentais (citadas acima) são representadas como fases separadas, tais como especificação de requisitos, projeto de software, implementação, teste, e assim por diante;
- Desenvolvimento incremental: intercala as atividades de especificação, desenvolvimento e validação. O sistema é desenvolvido como uma série de versões (incrementos), com cada versão sendo adicionadas funções em relação à versão anterior;
- Engenharia de Software orientada ao reuso: baseia-se na existência de um número significativo de componentes reutilizáveis. O processo de desenvolvimento do software é centrado na integração desses componentes ao invés de desenvolvê-los do zero.

Pressman (2010) descreve um modelo genérico de processos, onde sua estrutura contém cinco atividades: comunicação, planejamento, modelagem, construção e implantação. E um conjunto de atividades de projeto: rastreamento e controle, gestão de risco, garantia de qualidade, gerenciamento de configuração, revisões técnicas, entre outros. Ele enquadra as atividades, ações e tarefas em quatro fluxos de processos distintos:

- Fluxo linear: executa cada uma das cinco atividades em sequência, iniciando com a comunicação e finalizando com a implantação;
- Fluxo iterativo: repete uma ou mais atividades antes de prosseguir para a atividade seguinte;
- Fluxo evolutivo: executa as atividades em espiral. Cada evolução através das cinco atividades, conduz a uma versão mais completa do software;
- Fluxo paralelo: executa uma ou mais atividades em paralelo com outras atividades.

A escolha do conjunto de tarefas, deve ser adaptada às necessidades do projeto e características da equipe do projeto (Pressman, 2010; Sommerville, 2011).

### 2.3 Manutenção de software

Manutenção de software é o processo geral de mudanças de um software após ter sido entregue. As mudanças podem ser simples como a correção de erros de codificação, ou mais extensas como a correção de erros de projeto, melhorias para corrigir erros de especificação, ou a implementação de novas exigências (Sommerville, 2011).

Além de softwares recém-entregues, as organizações precisam manter os sistemas legados, muitas vezes críticos para o negócio. Eles podem utilizar linguagens e tecnologias ultrapassadas, possuírem uma estrutura degradada por mudanças, e normalmente a documentação é insuficiente, apresentando grande risco para ser substituído (Sommerville, 2011). A rotatividade das pessoas, também afeta a manutenção de software. Quem desenvolveu o software pode não fazer mais parte da empresa, e então há falta de conhecimento para mantê-lo, o que pode comprometer na agilidade e eficácia do processo de manutenção (Pressman, 2010).

A manutenção de software, consome grande parte do custo total do ciclo de vida de um software, e isso não tem sofrido significativa mudança com o passar do tempo. A consultoria de TI Global Accenture (Huff; Glover; Hillman, 2014) afirma que o software está entre a primeira e segunda categoria, de mais gastos para a TI, onde

em média 55 por cento desses gastos, são utilizados para a manutenção de software. A *Forrester Research* realizou em 2013, entrevistas com líderes de TI em mais de 3.700 empresas e apontou que em média 72 por cento de seus orçamentos são utilizados para manter os softwares em funcionamento, e apenas 28 por cento é direcionado a novos projetos (Zetlin, 2013). Sneed (2008) e Glass (2001) destacam que a manutenção de software consome em média 60 por cento do esforço total gasto durante todo o ciclo de vida de um software.

## 2.4 Processo de manutenção de software

A manutenção do software se inicia com um desafio, pois as filas para correções de *bugs* são crescentes, assim como as solicitações de adaptação e melhorias. As organizações passam a ter mais gastos para manter os softwares em funcionamento, do que em novos projetos (Pressman, 2010; Glass, 2001). A mudança no software é inevitável, e por isso devem ser desenvolvidos mecanismos para avaliar, controlar e fazer as modificações.

A norma IEEE 14764-2006, voltada à gestão de atividades de manutenção de software, afirma que o ideal é que o planejamento para as atividades de manutenção se iniciem, durante a fase de planejamento do desenvolvimento do software (Croll; Pigoski; Moore, 2006). O software necessita essencialmente de uma alta capacidade evolutiva, e para isso é indispensável a utilização de padrões de projetos em sua construção, de modo a facilitar a sua compreensão, e a descoberta de potenciais problemas de manutenção antes de sua liberação (Pressman, 2010).

Definir uma estrutura e realizar uma classificação para cada solicitação de manutenção de software é essencial, pois possibilita a determinação de prioridades. A norma IEEE 14764-2006 (Croll; Pigoski; Moore, 2006), classifica as requisições de mudanças em correção e aprimoramento. E as classificações possuem os seguintes tipos de manutenção:

- Correção:
  - Manutenção corretiva: modificar o software corrigindo falhas descobertas após a entrega;

- Manutenção preventiva: sua finalidade é detectar e corrigir falhas que possam se transformar em problemas operacionais.
- Aprimoramento:
  - Manutenção adaptativa: modificar o software para mantê-lo em funcionamento em um ambiente diferente;
  - Manutenção perfectiva: modificar o software após a entrega, corrigindo falhas antes que sejam manifestadas como falhas.

Sommerville (2011), classifica a manutenção de software de forma semelhante a norma IEEE 14764-2006 (Croll; Pigoski; Moore, 2006), com exceção da manutenção preventiva.

- Reparação de falhas: reparo de erros referente a codificação, projeto ou requisitos;
- Adequação ao ambiente: o software deve ser modificado, para adaptá-lo a mudanças de ambiente como em hardware, plataforma operacional ou alteração em software de apoio;
- Funcionalidade adicional: mudanças necessárias quando os requisitos do sistema mudam devido à mudança organizacional ou de negócios. A escala desse tipo de mudança é a mais frequente.

A Figura 1 apresenta um valor médio que cada tipo de manutenção representa no esforço total despendido em manutenção. Onde para a reparação de falhas, o esforço é de 17 por cento, para adequação ao ambiente é de 18 por cento e para modificação ou funcionalidade adicional é de 65 por cento. Sommerville (2011) menciona que existem diferentes definições para os tipos de manutenção de software nas literaturas relacionadas ao assunto, e por isso é importante se atentar a possíveis divergências de contextos.

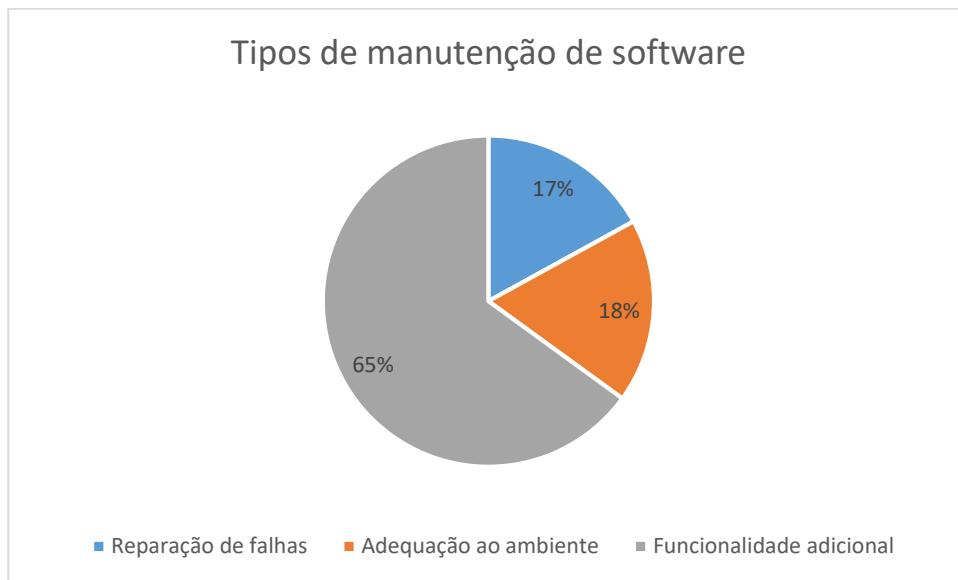


Figura 1 – Custo de cada tipo de manutenção

Fonte: Adaptado de Sommerville (2011)

A estrutura de atividades realizadas no processo de manutenção de software, proposta pela norma IEEE 14764-2006 (Croll; Pigoski; Moore, 2006) se divide em:

- Implementação do processo: o mantenedor estabelece os planos e procedimentos que serão necessários durante o processo de manutenção em paralelo aos planos de desenvolvimento;
- Análise de problemas e modificações: durante essa atividade o mantenedor analisa a solicitação de mudança, replica ou verifica o problema, desenvolve as opções para implementar a mudança, documenta a solicitação de mudança, os resultados e as opções de execução, obtém aprovação para a opção de mudança selecionada.
- Implementação da modificação: o mantenedor desenvolve e testa a modificação produzida no software;
- Revisão/Aceitação de Manutenção: esta atividade assegura que as modificações ao sistema estão corretas e que foram realizadas de acordo com as normas aprovadas utilizando a metodologia correta;
- Migração: durante a vida de um sistema, ele pode ter que ser modificado para ser executado em diferentes ambientes. Para migrar um sistema para um novo

ambiente, o mantenedor precisa determinar as ações necessárias para realizar a migração e, em seguida, desenvolver e documentar as etapas necessárias para efetuar a migração;

- Retirada do software: Uma vez que um produto de software atingiu o fim de sua vida útil, ele deve ser retirado. Uma análise deve ser realizada para ajudar a tomar a decisão de retirar um produto de software. A análise é muitas vezes de base econômica e pode ser incluído no Plano de Aposentadoria. A análise deve determinar se é rentável:
  - Reter tecnologia desatualizada;
  - Mudança para a nova tecnologia através do desenvolvimento de um novo produto de software;
  - Desenvolver um novo produto de software para atingir a modularidade;
  - Desenvolver um novo produto de software para facilitar a manutenção;
  - Desenvolver um novo produto de software para atingir a padronização;
  - Desenvolver um novo produto de software para facilitar a independência do fornecedor.

O produto de software pode ser substituído por um novo produto de software, mas em alguns casos não será substituído. A fim de aposentar um produto de software, o mantenedor deve determinar as ações necessárias para realizar a aposentadoria e, em seguida, desenvolver e documentar as etapas necessárias para efetuar a aposentadoria. Deve-se considerar o acesso a dados armazenados pelo software aposentado.

A norma não pressupõe o uso de nenhum modelo de desenvolvimento (cascata, espiral, entre outros). A sua utilização não é restringida pelo tamanho, complexidade, criticidade, ou aplicação do produto de software, e há um conjunto de critérios de aceitação para definir se as atividades estão em conformidade com a norma (Croll; Pigoski; Moore, 2006).

## 2.5 Melhoria de processos de software

A melhoria nos processos de software ou *Software Process Improvement* (SPI) atualmente, é uma das principais prioridades da indústria de TI (Basumatary; Baruah; Borgohain, 2014). Uma preocupação das organizações está em como melhorar a qualidade do software, e uma maneira de alcançar esse objetivo é melhorando os processos de software (Srivastava; Awasti, 2014). A SPI, visa melhorar os processos de software incluindo diversas atividades, como escopo, avaliação, projeto, implementação e melhoria contínua (Kuhrmann; Diebold; Munch, 2016). A estratégia SPI transforma a abordagem existente para o desenvolvimento de software, em algo mais focado, repetível e confiável em termos de qualidade do produto e prazo de entrega (Srivastava; Awasti, 2014).

Os programas de melhoria de processos são ainda mais importantes quando considerado todo o cenário de desenvolvimento de projetos de software que falharam parcial ou totalmente, os numerosos incidentes e as perdas financeiras geradas por essas falhas (Laporte; O'Connor, 2015).

### 2.5.1 Modelos de processos de melhoria

A melhoria de processos de software é reconhecida como um importante fator de sucesso para as organizações que operam em uma economia competitiva, pois o resultado de sua aplicação pode levar a um aumento na velocidade de desenvolvimento, na qualidade e confiabilidade do produto. Existem inúmeras abordagens padronizadas de SPI, para apoiar os engenheiros de processos em sua gestão (Kuhrmann; Beecham, 2014).

Grande parte dos modelos de processos descreve uma abordagem para a melhoria dos mesmos (Srivastava; Awasti, 2014). A seguir, o exemplo de duas abordagens de SPI, suas descrições e principais características.

## A. CMMI

O CMMI é um modelo de melhoria continua de processos, desenvolvido pelo *Software Engineering Institute* (SEI). O modelo é baseado nas melhores práticas de desenvolvimento e manutenção de software, com o objetivo de reduzir e refinar os processos de forma a simplificá-los, evitando desperdícios e alcançando progressos reais de desenvolvimento para a organização (Basumatary; Baruah; Borgohain, 2014). O modelo é apresentado em cinco níveis de maturidade, para que seja possível e transparente, saber o que é necessário para progredir de um nível para o outro (Srivastava; Awasti, 2014). Além dos níveis, o modelo apresenta um conjunto de áreas de processos que são agrupadas em categorias. Dessa forma, é possível que a empresa mantenha maior foco na área ou categoria de sua necessidade (SEI, 2010).

A Tabela 1 apresenta os níveis de processos, suas áreas de processos e categorias.

Tabela 1 – Modelo CMMI

Nível de maturidade	Categoria	Área de processo
Nível 5 - Em otimização	Suporte	Análise e resolução de causas
	Gestão de processo	Implantação de inovações na organização
Nível 4 - Gerenciado quantitativamente	Gestão de processo	Desempenho dos processos da organização
	Gestão quantitativa de projeto	Gestão Quantitativa de Projeto
Nível 3 - Definido	Suporte	Análise e Tomada de Decisões
	Gestão de processo	Definição dos Processos da organização
	Gestão de processo	Foco nos Processos da organização
	Gestão de processo	Treinamento na organização
	Engenharia	Integração de produto
	Engenharia	Desenvolvimento de requisitos
	Gestão de projeto	Gestão de riscos
	Engenharia	Solução técnica
	Engenharia	Validação
	Engenharia	Verificação
Nível 2 - Gerenciado	Suporte	Medição e análise
	Suporte	Gestão de configuração
	Gestão de projeto	Monitoramento e controle de projeto
	Gestão de projeto	Planejamento de projeto
	Suporte	Garantia da qualidade de processo e produto
	Engenharia	Gestão de requisitos
	Gestão de projeto	Gestão de contrato com fornecedores
Nível 1 - Executado		Inicial

(Adaptado de SEI, 2010)

Abaixo uma breve descrição de cada nível de maturidade do modelo CMMI (SEI, 2010):

- Nível 1 - Inicial: os processos normalmente não possuem padrão, ou estão no caos;
- Nível 2 - Gerenciado: os projetos possuem seus requisitos. Existe planejamento, medição e controle dos processos;
- Nível 3 - Definido: os processos possuem um padrão, estão claramente definidos e compreendidos para a organização;
- Nível 4 - Gerenciado Quantitativamente: ocorre o aumento da previsibilidade do desempenho de diferentes processos, uma vez que os mesmos já são controlados quantitativamente;
- Nível 5 - Otimizado: a organização tem os meios para identificar seus mais fracos elementos do processo e fortalecê-los através da definição de melhorias.

O CMMI considera três dimensões críticas, onde normalmente as organizações se concentram: pessoas, procedimentos e métodos, e ferramentas e equipamentos. E considera que o processo utilizado na empresa, é o que mantém a coesão dessas dimensões. Com a adoção de processos é possível otimizar os recursos, e obter uma melhor compreensão das tendências de negócio (SEI, 2010).

## **B. MPS.BR**

O modelo MPS.BR, coordenado pela Softex, é baseado nas melhores práticas de Engenharia de Software, de acordo com os padrões internacionais de produção. O seu foco é na melhoria dos processos de softwares da indústria de software brasileira, com especial atenção a micro, pequenas e médias empresas. O MPS.BR possui sete níveis de maturidade (Tabela 2), e os processos são definidos em termos de propósitos e resultados, que seriam os objetivos gerais e os resultados esperados. Para se atingir os resultados esperados, há um conjunto de atributos do processo a serem alcançados, que indicam uma característica mensurável da capacidade do processo, sendo aplicável a qualquer processo. A escala de maturidade progride do

nível inicial ao nível final através de um grupo de processos existente em cada nível, que indica onde a organização deve colocar o esforço da melhoria. E o progresso do nível ocorre quando os resultados de cada processo são alcançados (MPS.BR, 2016).

Tabela 2 – Modelo MR-MPS-SW

Nível de maturidade	Processo
A - Em otimização	Otimização
B - Gerenciado quantitativamente	Gerência de projetos (evolução)
C - Definido	Gerência de riscos
	Desenvolvimento para reutilização
D - Largamente definido	Gerência de decisões
	Verificação
	Validação
	Projeto e construção do produto
	Integração do produto
	Desenvolvimento de requisitos
E - Parcialmente definido	Gerência de projetos (evolução)
	Gerência de reutilização
	Gerência de recursos humanos
	Definição de processo organizacional
	Avaliação e melhoria do processo organizacional
F - Gerenciado	Medição
	Garantia de qualidade
	Gerência de portfólio de projetos
	Gerência de configuração
	Aquisição
G - Parcialmente gerenciado	Gerência de requisitos
	Gerência de projetos

(Adaptado de MPS.BR, 2016)

Segue uma breve descrição de cada nível de maturidade do modelo (MPS.BR, 2016):

- A – Em Otimização: contempla o processo de melhorias incrementais, inovações, e otimização continua. São identificadas as mudanças necessárias no processo a partir da análise de defeitos e problemas;
- B – Gerenciado Quantitativamente: avalia-se o desempenho dos processos através de métricas;

- C – Definido: é realizado o gerenciamento de riscos;
- D – Largamente Definido: envolve verificação, validação, além da liberação, instalação e integração de produtos, dentre outras atividades;
- E – Parcialmente Definido: considera processos como treinamento, adaptação de processos para gerência de projetos, além da preocupação com a melhoria e o controle do processo organizacional;
- F – Gerenciado: introduz controles de medição, gerência de configuração, conceitos sobre aquisição e garantia da qualidade;
- G – Parcialmente Gerenciado: neste ponto deve-se iniciar o gerenciamento de requisitos e do projeto.

Entre as bases técnicas para a definição do MPS, estão os modelos *Capability Maturity Model Integration for Services* (CMMI-SVC), focado nas melhores práticas de melhoria de processos para empresas prestadoras de serviços de TI, e o *People CMM* que contribui para minimizar e, por vezes, eliminar os problemas críticos relativos a gestão de pessoas (MPS.BR, 2016).

Como o estudo de caso aqui proposto envolve a aplicação do processo Gerência de Projetos do MPS.BR nível G, este processo será detalhado na próxima seção.

### **Nível G – Parcialmente Gerenciado**

Para a aplicação do modelo na organização estudada, considerou-se o nível de maturidade G - Parcialmente Gerenciado, que é o primeiro nível para a aplicação do modelo MPS-BR. Este nível compreende dois processos (MPS.BR, 2016): Gerência de Projetos e Gerência de requisitos. Para a aplicação no estudo de caso, será utilizado o processo Gerência de Projetos.

A implementação do processo deve satisfazer os seguintes atributos:

- AP 1.1 O processo é executado: é a medida do quanto o propósito do processo é alcançado pela sua execução. O resultado da implementação deste atributo de processo é:
  - O processo produz os resultados definidos.
- AP 2.1: A execução do processo é gerenciada: é a medida do quanto a execução do processo é gerenciada. Os resultados da implementação deste atributo de processo são:
  - Existe uma política organizacional estabelecida e mantida para o processo;
  - A execução do processo é planejada (O planejamento deve incluir identificação e disponibilização dos recursos e informações necessárias para a execução do processo, definição, atribuição e comunicação das responsabilidades pela execução do processo e planejamento da comunicação entre as partes interessadas);
  - A execução do processo é monitorada em relação ao planejado e, quando necessário, ajustes são realizados;
  - As pessoas que executam o processo estão preparadas para executar suas responsabilidades;
  - As atividades, o status e os resultados do processo são revistos com a gerência de nível superior e são tratadas questões críticas;
  - (A partir do Nível F) A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente e são tratadas as não conformidades.

Os propósitos e resultados esperados para o processo Gerência de Projetos são:

- Propósito

Estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver

desvios significativos no desempenho do projeto. O propósito deste processo evolui à medida que a organização cresce em maturidade. Assim, a partir do nível E, alguns resultados evoluem e outros são incorporados, de forma que a gerência de projetos passe a ser realizada com base no processo definido para o projeto e nos planos integrados. No nível B, a gerência de projetos passa a ter um enfoque quantitativo, refletindo a alta maturidade que se espera da organização. Novamente, alguns resultados evoluem e outros são incorporados.

- Resultados esperados:

- GPR 1. O escopo do trabalho para o projeto é definido;
- GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
- GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos;
- GPR 4. (Até o nível F) O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;
- GPR 4. (A partir do nível E) O planejamento e as estimativas das tarefas do projeto são feitos baseados no repositório de estimativas e no conjunto de ativos de processo organizacional;
- GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;
- GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
- GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
- GPR 8. (Até o nível F) Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;

- GPR 8. (A partir do nível E) Os recursos e o ambiente de trabalho necessários para executar os projetos são planejados a partir dos ambientes padrão de trabalho da organização;
- GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
- GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;
- GPR 11. A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados;
- GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido;
- GPR 13. O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado;
- GPR 14. Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado;
- GPR 15. Os riscos são monitorados em relação ao planejado;
- GPR 16. O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido;
- GPR 17. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;
- GPR 18. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
- GPR 19. Ações para corrigir desvios em relação ao planejado e para prevenir repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;

- GPR 20. (A partir do nível E) Equipes envolvidas no projeto são estabelecidas e mantidas a partir das regras e diretrizes para estruturação, formação e atuação;
- GPR 21. (A partir do nível E) Experiências relacionadas aos processos contribuem para os ativos de processo organizacional;
- GPR 22. (A partir do nível E) Um processo definido para o projeto é estabelecido de acordo com a estratégia para adaptação do processo da organização;
- GPR 22. (A partir do nível B) Os objetivos de qualidade e de desempenho do processo definido para o projeto são estabelecidos e mantidos;
- GPR 23. (A partir do nível B) O processo definido para o projeto que o possibilita atender seus objetivos de qualidade e de desempenho é composto com base em técnicas estatísticas e outras técnicas quantitativas;
- GPR 24. (A partir do nível B) Subprocessos e atributos críticos para avaliar o desempenho e que estão relacionados ao alcance dos objetivos de qualidade e de desempenho do processo do projeto são selecionados;
- GPR 25. (A partir do nível B) Selecionar medidas e técnicas analíticas a serem utilizadas na gerência quantitativa;
- GPR 26. (A partir do nível B) O desempenho dos subprocessos escolhidos para gerência quantitativa é monitorado usando técnicas estatísticas e outras técnicas quantitativas;
- GPR 27. (A partir do nível B) O projeto é gerenciado usando técnicas estatísticas e outras técnicas quantitativas para determinar se seus objetivos de qualidade e de desempenho do processo serão atingidos;
- GPR 28. (A partir do nível B) Questões que afetam os objetivos de qualidade e de desempenho do processo do projeto são alvo de análise de causa raiz.

### 2.5.2 Problemas na implantação de melhoria de processos

A implementação de melhoria de processos de software (SPI) nas organizações leva tempo, muitas vezes vários anos, para melhorar permanentemente a produtividade de seus esforços de desenvolvimento de software. São atividades incrementais realizadas a longo prazo, que envolvem a análise de processos, padronização, mensuração e mudança. Ao analisar, por exemplo, os aspectos de um mapa de processo de software para os níveis de maturidade CMM, é possível ter uma ideia aproximada da ordem em que a organização deve introduzir os processos para a sua equipe. As organizações que tentam fazer mudanças imediatas nos processos, em grande escala, tendem a fazer o modelo falhar (Srivastava; Awasti, 2014).

Pressman (1996) destaca que as organizações muitas vezes, visam a SPI como um caminho mágico, onde uma única estratégia garante níveis de maturidade mais elevados. O problema de implementar uma melhoria no processo de software, está no ambiente em que será implementado, pois a SPI em si, pode ser muito simples, afirma Pressman, que em seus 15 anos atuando com melhoria de processos de software, presenciou mudanças pequenas relacionadas em como as empresas abordam a SPI, e identificou algumas realidades nesse contexto:

- A gestão quer melhoria imediata do processo de software, mas não disponibiliza os recursos o tempo suficiente, falha na definição de metas, e não define prioridade para os objetivos que foram estabelecidos;
- Se a gestão não compromete os recursos, as pessoas tendem a se concentrar em coisas erradas ou inúmeras coisas, e não se mantém focada o tempo suficiente para a realização das atividades;
- Se as pessoas que vão desempenhar o trabalho de SPI, não possuem experiência de boas práticas de engenharia de software, elas não conseguirão aprender o suficiente sobre SPI, para tomar decisões e estabelecer as áreas chave do processo;
- Se a equipe está disponível, comprometida, e é razoavelmente experiente, a SPI é conduzida com resultados previsivelmente pobres, pois para que seja

possível prever resultados positivos, é imprescindível que os recursos sejam altamente experientes.

Os principais objetivos da SPI são entender os processos existentes, introduzir mudanças no processo para melhorar a qualidade e reduzir os custos ou acelerar o tempo de realização das atividades. Porém, o trabalho de melhoria é muitas vezes centrado na redução e prevenção de defeitos, afim de melhorar a qualidade do produto, enquanto há outros atributos de processo a serem considerados com atenção, como o ambiente de desenvolvimento da SPI, a gestão organizacional e de pessoas (Srivastava; Awasti, 2014).

### 2.5.3 O papel das pessoas

A implementação de melhorias de processo de software, é uma mudança muito particular para cada organização, e como qualquer mudança, os recursos humanos são considerados peça chave para uma implementação bem sucedida. As pessoas são os membros envolvidos na definição e execução de novos processos. Elas necessitam se adaptar, apoiar e dar suporte as mudanças continuamente (Matturo; Saavedra, 2012).

Para realizar a implantação de melhorias de processos, é preciso identificar os fatores de sucesso e as barreiras do ambiente a ser melhorado, afim de propor ações para gerenciar e definir as mudanças necessárias para a organização (Matturo; Saavedra, 2012). O *People CMM* patrocinado pelo SEI é um conjunto de boas práticas de gestão de pessoas, que fornece um roteiro para melhorar continuamente a capacidade da força de trabalho de uma organização. A capacidade dos trabalhadores pode ser definida como o nível de conhecimento, competência e habilidade nos processos disponíveis, para a realização de atividades. E indica fatores como:

- Prontidão para a realização de atividades críticas de negócio;
- Resultados previsíveis ao realizar essas atividades de negócios;
- Potencial para investir em melhoria de processo ou tecnologia avançada.

O *People CMM* descreve um caminho de melhoria, realizando mudanças nas práticas inconsistentes da força de trabalho, para uma infraestrutura madura, de forma a elevar continuamente a capacidade da força de trabalho da organização. A filosofia do *People CMM*, pode ser resumida em dez princípios conforme a Tabela 3 (Curtis; Hefley; Miller, 2009).

Tabela 3 – Filosofia do People CMM

Príncípio	Descrição
1	Nas organizações maduras, a capacidade da força de trabalho está diretamente relacionada ao desempenho do negócio.
2	A capacidade da força de trabalho é uma questão competitiva e uma fonte de vantagem estratégica.
3	A capacidade da força de trabalho deve ser definida em relação aos objetivos estratégicos de negócios da organização.
4	O trabalho intensivo em conhecimento, leva o foco dos elementos do trabalho para as competências da força de trabalho.
5	A capacidade pode ser medida e melhorada em vários níveis da organização, incluindo indivíduos, grupos de trabalho, competências da força de trabalho e organização.
6	Uma organização deve investir em melhorar a capacidade das competências da força de trabalho que são críticas, para a sua competência básica como um negócio.
7	A gestão operacional é responsável pela capacidade da força de trabalho.
8	A melhoria da capacidade da força de trabalho pode ser buscada como um processo composto por práticas e procedimentos comprovados.
9	A organização é responsável por fornecer oportunidades de melhoria, e os indivíduos são responsáveis para tirar proveito delas.
10	Como as tecnologias e as formas organizacionais evoluem rapidamente, as organizações devem evoluir continuamente suas práticas de força de trabalho e desenvolver novas competências.

(Adaptado de Curtis; Hefley; Miller, 2009)

O *People CMM* é um quadro evolutivo que orienta as organizações na seleção de prioridades das ações de melhoria, com base na maturidade atual das suas práticas de trabalho. Cada nível de maturidade representa uma escala evolutiva de novas capacidades para os indivíduos e para a organização. A natureza da transformação imposta pelas práticas da força de trabalho da organização para alcançar cada nível está representada na Figura 2.

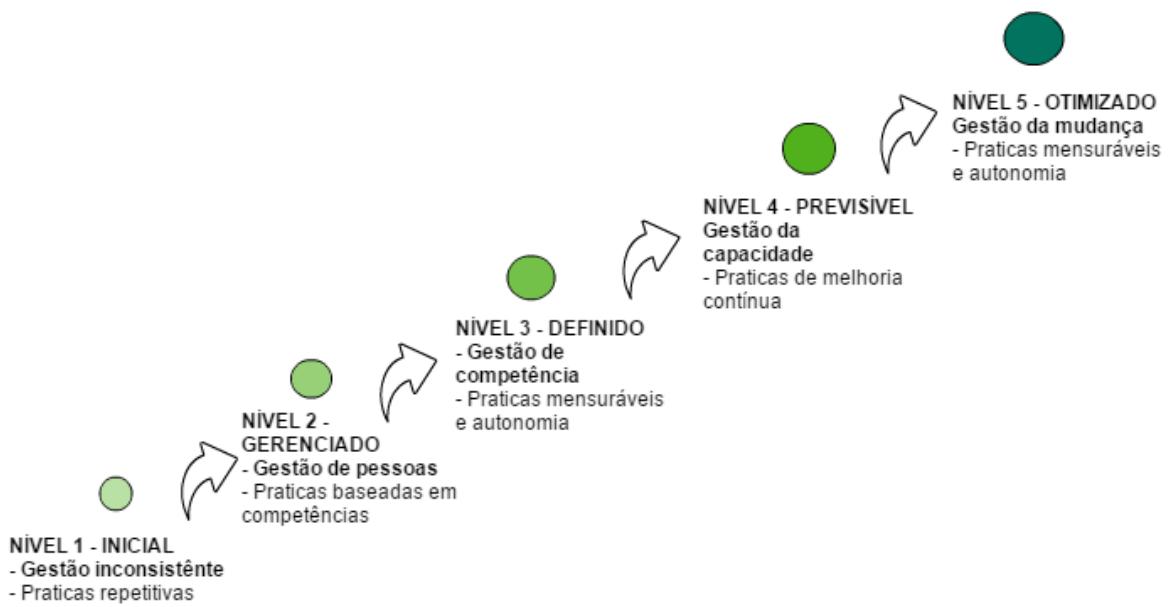


Figura 2 – Níveis de maturidade do People CMM

Fonte: Adaptado de Curtis, Hefley e Miller (2009)

O *People CMM* implanta práticas de força de trabalho cada vez mais sofisticadas a cada nível de maturidade. Com exceção do Nível Inicial, cada nível é caracterizado por um conjunto de práticas inter-relacionadas em áreas críticas da gestão da força de trabalho. Quando institucionalizadas e executadas adequadamente, essas práticas criam capacidades para a organização gerir e desenvolver a sua força de trabalho (Curtis; Hefley; Miller, 2009).

A Tabela 4 apresenta um resumo de cada nível de maturidade proposto pelo *People CMM*:

Tabela 4 – Modelo People CMM

Nível	Enfoque	Práticas	Objetivos	Processos
Nível 1 - Inicial	Não há	Não há	Não há	Não há
Nível 2 - Gerenciado	Gerenciamento de pessoas	Práticas repetidas	Gerenciamento e desenvolvimento da equipe pelos gerentes	Recrutamento; Comunicação e coordenação; Ambiente de trabalho; Gestão do desempenho; Treinamento e desenvolvimento; Remuneração.
Nível 3 - Definido	Gerenciamento de competências	Práticas baseadas em competências	Desenvolvimento de competências e grupos de trabalho, alinhando-os aos objetivos e estratégias do negócio	Análise de competências; Planejamento da força de trabalho; Desenvolvimento de competências; Desenvolvimento de carreira; Práticas baseadas em competências; Desenvolvimento de grupos de trabalho; Cultura participativa.
Nível 4 - Previsível	Gerenciamento das capacidades da empresa	Práticas mensuradas	Integração das competências no trabalho e gerenciamento quantitativo do desempenho	Integração de competências; Grupos de trabalho autônomos; Ativos baseados em competências; Gestão quantitativa de desempenho; Gestão da capacidade organizacional; Orientação e aconselhamento.
Nível 5 - Otimizado	Gerenciamento das mudanças na empresa	Práticas baseadas em melhorias contínuas	Instauração da melhoria contínua dos processos e alinhamento do grupo de trabalho com a capacidade organizacional	Melhoria contínua da capacidade; Alinhamento do desempenho organizacional; Inovação contínua da força de trabalho.

(Silveira; Guimarães; Abraão, 2007)

O principal objetivo do *People CMM* é a preparação das pessoas para as mudanças organizacionais, afim de mitigar os efeitos das barreiras, resistência e desmotivação, além de orientar, planejar e mensurar a efetiva aplicação das mesmas dentro da organização. O modelo é uma importante base comprovada de que o sucesso do projeto está relacionado ao envolvimento e preparação das pessoas (Matturo; Saavedra, 2012).

## 2.6 Qualidade de software

As falhas no software ou por quem opera o software, é a grande causa de grandes perdas financeiras nas organizações e por essa razão muitas pesquisas são realizadas em torno da qualidade no desenvolvimento de software.

Um software de má qualidade pode ser definido como um software defeituoso que não cumpre os requisitos dos clientes. As organizações lutam contra o tempo, para entregar software de qualidade e ficar à frente com as novas tecnologias. Há uma imensa pressão para que a entrega de um novo software ou um pacote de mudanças seja livre de *bugs* (Mohan; Shakar; Jayasridevi, 2012). A capacidade das organizações para competir, adaptar e sobreviver depende cada vez mais da qualidade, entre outros aspectos como produtividade, tempo de desenvolvimento e custos. E mesmo assim, especialistas em software gastam em média 45 por cento de seu tempo em evitáveis retrabalhos (Laporte; O'Connor, 2015).

Existe uma grande importância em se avaliar a qualidade de um software e mesmo que ele tenha sido entregue no prazo, dentro do orçamento e de acordo com as especificações, isso não pressupõe que ele não possua problemas. Problemas comuns podem ser citados, tais como:

- O software pode ser difícil de entender e modificar, levando a custos excessivos e não triviais em sua manutenção;
- O software pode ser difícil de usar ou de fácil uso indevido;
- O software pode ser de difícil integração com outros sistemas, o que se torna mais problemático a cada dia, pois inúmeros outros programas são criados.

Pode-se considerar, que uma das primeiras recompensas do aumento da capacidade de lidar com a qualidade de software, é a melhoria em sua manutenção relacionada a custo-eficácia. A qualidade não se traduz em altos custos com manutenção, ou seja, altos custos com correções de erros e / ou resposta a novas necessidades durante o ciclo de vida do software (Boehm; Brown; Lipow, 1976).

Há processos de garantia de qualidade, como por exemplo *Software Quality Assurance* (SQA), que define um conjunto de atividades que apoiam na produção de

softwares que atendam as especificações e uso pretendido. Organizações que possuem processos focados na garantia da qualidade, tendem a obter melhores cenários com relação as que não possuem: o custo com manutenção diminui, ocorre uma melhoria na qualidade do produto, o processo é continuamente melhorado e um acompanhamento é realizado em torno das falhas e indicadores de qualidade (Lewis, 2004).

## 2.7 Fatores de sucesso para projetos de software

Uma pesquisa divulgada pela *Standish Group* (Lynch, 2015), chamada *2015 Chaos Report*, estudou 50 mil projetos de desenvolvimento de software em todo o mundo, desde pequenos aprimoramentos até implementações de reengenharia, e apontou que em 2015, 29 por cento dos projetos obtiveram o sucesso esperado, 52 por cento foram concluídos com algum atraso, problemas no orçamento ou menos funcionalidades que o previsto, e 19 por cento não foram concluídos, pois falharam completamente. A Tabela 5 apresenta os resultados dos projetos de software referente a um período de cinco anos (2011 a 2015).

Tabela 5 - Resultado dos projetos de software de 2011 a 2015

Projetos	2011	2012	2013	2014	2015
Sucesso	29%	27%	31%	28%	29%
Desafio	49%	56%	50%	55%	52%
Falhou	22%	17%	19%	17%	19%

(Adaptado de Lynch, 2015)

Foi possível comparar os resultados obtidos de 2011 a 2015, referente à utilização de metodologias ágeis e tradicionais, onde projetos que utilizaram a abordagem ágil resultaram em entregas mais bem sucedidos, conforme aponta a Tabela 6.

Tabela 6 - Método Ágil X Cascata de 2011 a 2015

Método	Sucesso	Desafio	Falhou
Ágil	39%	52%	9%
Cascata	11%	60%	29%

(Adaptado de Lynch, 2015)

O estudo reuniu fatores, que ao longo dos anos de pesquisa, em conjunto, podem tornar os projetos de software mais bem sucedidos. Segue a descrição para alguns fatores relacionados:

- Apoio executivo: é quando um executivo ou grupo de executivos concorda em fornecer apoio financeiro e emocional. Os executivos encorajarão e ajudarão na conclusão bem-sucedida do projeto;
- Maturidade emocional: é a coleção de comportamentos básicos de como as pessoas trabalham juntas;
- Envolvimento do usuário: ocorre quando os usuários estão envolvidos no processo de tomada de decisão e coleta de informações. O que inclui comentários dos usuários, revisão de requisitos, pesquisa básica, prototipagem e outras ferramentas;
- Funcionários qualificados: são pessoas que entendem do negócio e da tecnologia. Um pessoal qualificado é altamente proficiente na execução dos requisitos do projeto e entrega do projeto ou produto;
- Proficiência ágil: significa que a equipe ágil e o proprietário do produto são hábeis no processo ágil. A proficiência ágil é a diferença entre bons resultados e desfechos ruins com a utilização de métodos ágeis;
- Conhecimentos de gerenciamento de projetos: é a aplicação de conhecimentos, habilidades e técnicas às atividades do projeto, afim de atender ou exceder as expectativas dos *stakeholders* e produzir valor para a organização;

- Claros objetivos de negócios: é a compreensão de todos os interessados e participantes do objetivo comercial para a execução do projeto. E pode significar também, que o projeto está alinhado com os objetivos da organização e estratégia.

## 2.8 Considerações finais do capítulo

O desenvolvimento de software é uma tarefa complexa e depende de inúmeros fatores, como processos, pessoas, gestão, qualidade, tempo, custo, entre outros, para um resultado bem sucedido. E ainda assim, um resultado bem sucedido na entrega do produto final ao cliente, não elimina a possibilidade de que o software possua problemas de manutenção ao longo do seu ciclo de vida.

Os problemas com a manutenção do software possuem diferentes naturezas e por isso a importância da utilização de boas práticas de engenharia de software, desde a concepção do software até o processo de extensão do seu ciclo de vida.

Nesse trabalho é utilizado como base o modelo de melhoria contínua de processos MPS.BR, por se focar na implantação de melhoria contínua de processos para indústrias de software brasileiro. Diferente do CMMI, que possui 5 níveis, o MPS.BR coloca sete níveis de maturidade, com o objetivo de possibilitar uma implementação e avaliação adequada a micros, pequenas e médias empresas, facilitando a passagem de um nível para outro. Ao adotar o MPS.BR, a empresa poderá chegar a um nível inicial de maturidade e capacidade, com um grau menor de esforço e de investimento (Bertolini; Silveira; Ceratti, 2016).

### 3 DESENVOLVIMENTO

#### 3.1 Considerações iniciais

O mercado financeiro foi um dos primeiros a experimentar a força da globalização através de softwares bancários (Pavel; McElravey, 1990). Ao longo dos anos os sistemas de software se tornaram vitais para as operações, e garantir sua disponibilidade e evolução contínua é fundamental para assegurar a continuidade dos negócios.

A organização estudada possui cerca de 30 anos no mercado de investimento e em torno de 2 mil funcionários, além de mão de obra terceirizada presente em praticamente todas as áreas de TI. Os seus clientes se dividem em pessoas físicas de alto patrimônio, pessoas jurídicas, governo, instituições financeiras, entre outros. Sua sede fica no Brasil e possui escritórios entre o território nacional e internacional.

Suas principais áreas de atuação são:

- Administração de fundos: a organização operacionaliza o fundo de investimento do cliente, e recebe uma taxa para isso;
- Empréstimos: o cliente faz um contrato de um valor com a organização, que lhe concede o valor, e o cliente tem um prazo determinado para devolver o valor, com o acréscimo de juros e encargos;
- Financiamentos: compra parcelada de um produto ou serviço, com associação de juros ao valor inicial;
- Assessoria financeira: profissionais capacitados auxiliam nas tomadas de decisões de assuntos financeiros, como por exemplo na definição de qual é o melhor investimento para uma família ou organização;
- Gestão de ativos: é a gestão dos custos e riscos ao longo do ciclo de vida do ativo, que podem ser equipamentos, contratos, ferramentas;
- Investimentos imobiliários: aplicação de recursos em qualquer tipo de negócio com base imobiliária (prédios comerciais, imóveis prontos, empreendimentos, entre outros);
- Câmbio: compra e venda de moedas.

A área de manutenção de software da organização passou por uma significativa reestruturação a partir de janeiro de 2015. A mudança iniciou com a troca de recursos em todos os níveis (líderes, gestores e analistas). E se estendeu aos processos, pois como não eram utilizados adequadamente e melhorados continuamente, não era possível medir os resultados da área como um todo. Os principais fatores motivadores da mudança foram: a desorganização e falta de controle da gestão, resistência das pessoas em utilizar os processos, desmotivação dos recursos, reclamações recorrentes das áreas de negócio quanto ao atendimento realizado por equipes de manutenção de software, e a falta de qualidade no atendimento aos chamados.

### 3.2 Áreas de Tecnologia da Informação da organização

O foco do estudo de caso está relacionado à área de manutenção de software, porém se faz necessário entender de forma sucinta o papel de outras áreas de TI, para que seja possível visualizar a integração e dependência que o processo de manutenção de software tem com as demais áreas. As principais áreas de TI da organização são: atendimento 1, projeto, manutenção de software, infraestrutura e banco de dados. Embora a área de manutenção de software tenha integração com outras áreas, como por exemplo: redes e *middleware*, as áreas mais acionadas são infraestrutura e banco de dados, por isso ambas estão presentes entre as principais áreas de TI.

A área de manutenção de software possui diferentes equipes, de acordo com a área de negócio a qual realiza atendimento. As áreas de negócio são as áreas de usuários dos sistemas de atendimento. Cada área de negócio está associada a um segmento da organização.

As nomenclaturas a seguir são utilizadas ao longo deste trabalho. Segue uma breve descrição para cada uma:

- Segmentos: são as áreas de atuação da organização, onde uma ou mais áreas fazem parte de um segmento: crédito (empréstimos e financiamentos), gestão

de ativos, investimentos (assessoria financeira, administração de fundos e investimentos imobiliários) e câmbio;

- Sistemas de atendimento: são os softwares desenvolvidos internamente ou por fornecedores, utilizados pelas áreas de negócio e de responsabilidade das equipes de manutenção de software e projeto. Um software pode ser utilizado por uma ou mais áreas de negócio, através de módulos distintos, logo um software pode ser da responsabilidade de uma ou mais equipes de manutenção/projeto;
- Sistema de demandas (SD): É utilizado por todas as áreas do banco, para o cadastro e acompanhamento de solicitações de algo pertinente a outra área / equipe;
- Sistema de Melhorias (SM): As solicitações de melhorias são cadastradas, priorizadas e acompanhadas através do SM. Utilizado exclusivamente por equipes de manutenção de software.

O papel de cada área de TI da organização é:

- Atendimento 1: responsável pelo cadastro de demandas no SD, assim como a análise inicial das solicitações para a sua correta classificação e categorização no sistema;
- Projeto: atua no desenvolvimento de novos softwares, na implementação de novas necessidades para softwares existentes e em melhorias cujo tempo de realização, seja estimado em mais de 40 horas;
- Manutenção de software: realiza a manutenção nos softwares, incluindo melhorias com tempo de realização estimado em menos de 40 horas, correção de *bugs*, além de esclarecimento de dúvidas e extração de dados;
- Infraestrutura: realiza atividades relacionadas a solicitações de alterações de versões de softwares em produção (pacotes, executáveis, arquivos, configurações, etc.), redes, telecomunicação, sistema operacional, hardware, etc.;

- Banco de dados: realiza solicitações de execuções de scripts nas bases de dados. Controlam os acessos e permissões, o estado e as atualizações das ferramentas de Sistema de Gerenciamento de Banco de Dados (SGBD).

A Figura 3 é uma apresentação em alto nível, da integração entre as áreas de TI, áreas de negócio e sistemas de atendimento, com destaque na área de manutenção de software.

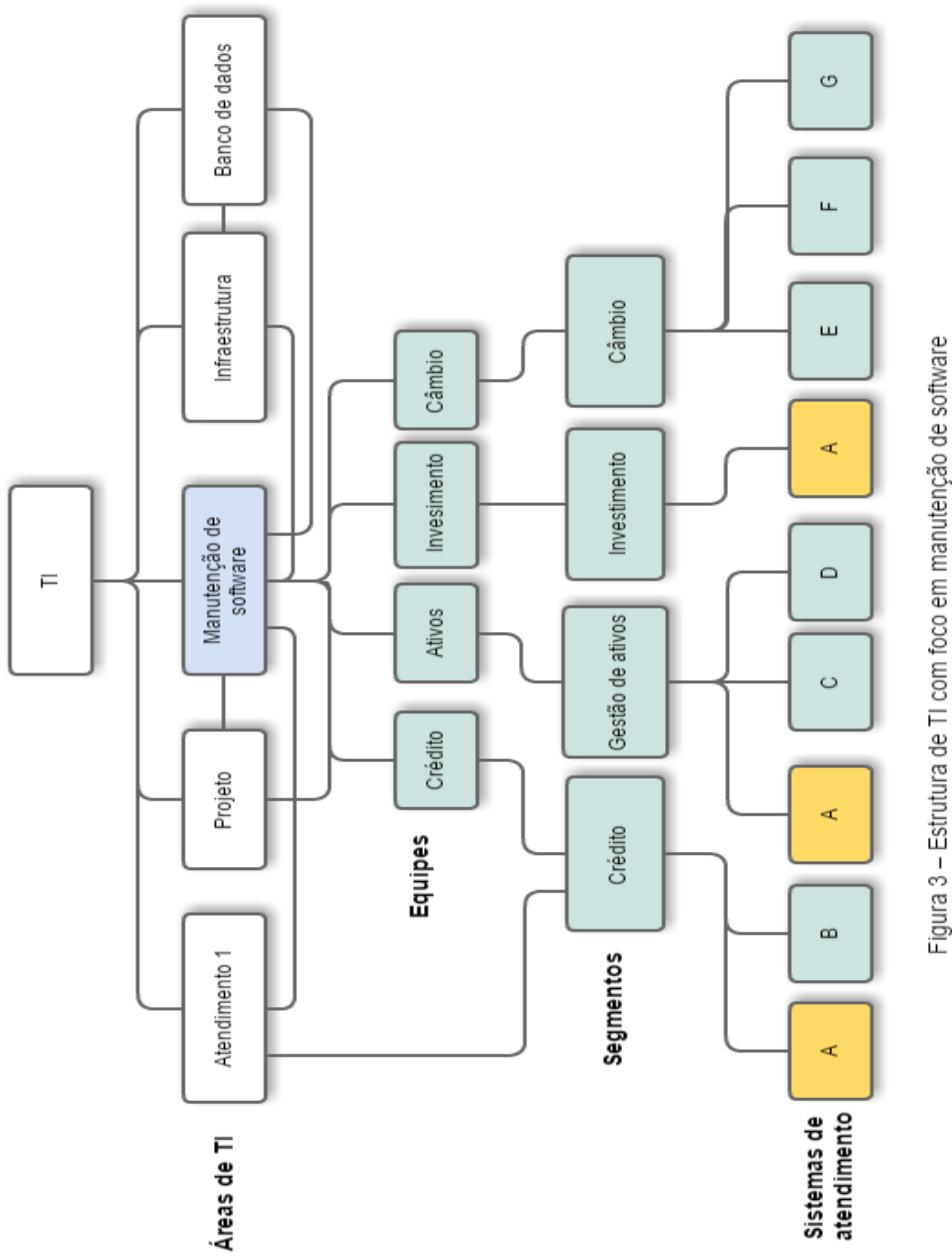


Figura 3 – Estrutura de TI com foco em manutenção de software

### 3.3 Área de manutenção de software da organização

#### 3.3.1 Equipe de manutenção de software

A organização possui cerca de 45 profissionais, entre funcionários e recursos terceirizados, dedicados as atividades de manutenção de software. A área de manutenção de software é dividida em equipes. Cada equipe é dedicada a uma área de negócios, que pode utilizar um ou mais sistemas de atendimento. O nível de carreira dos profissionais de manutenção se classifica entre júnior, pleno, sênior e máster, de acordo com o tempo de carreira e experiência, sendo que a maioria dos recursos é de nível pleno e sênior. É importante para a organização que seus recursos sejam capacitados, devido a constante necessidade de ações imediatas para correção ou contorno de problemas de alta complexidade encontrados nos sistemas de atendimento assim como, a realização de melhorias nos sistemas utilizando boas práticas de Engenharia de Software.

Uma associação pode ser realizada referente à capacitação exigida aos profissionais da área, conforme os fatores citados pelo modelo de gestão de pessoas *People-CMM* (Curtis; Hefley; Miller, 2009):

- Prontidão para a realização de atividades críticas de negócio;
- Resultados previsíveis ao realizar essas atividades de negócios;
- Potencial para investir em melhoria de processo ou tecnologia avançada.

O efetivo acompanhamento e monitoramento das filas de *backlog* dos sistemas de atendimento, também é de responsabilidades dos profissionais de manutenção de software. Através dos resultados de relatórios extraídos do SD, a gestão pode medir, controlar e identificar as mudanças necessárias para os processos e para a equipe.

### 3.4 Processo de manutenção de software

Estabelecer, monitorar e melhorar continuamente os processos de manutenção de software é um desafio constante para que a gestão possa acompanhar de maneira efetiva a atuação e os resultados de cada equipe, identificar as barreiras, promover mudanças, treinamentos, avaliar a necessidade dos recursos e medir o desempenho de cada profissional.

Cerca de 90 sistemas de atendimento, sejam eles desenvolvidos por fornecedores ou internamente, são mantidos pela área de manutenção de software. Os sistemas são na maioria legados, foram desenvolvidos em diferentes linguagens e tecnologias, e normalmente não é possível identificar um padrão de desenvolvimento, a documentação foi perdida, é escassa ou está desatualizada, e quem o desenvolveu não faz mais parte da organização. Porém, eles são extremamente importantes para o negócio, fornecem serviços essenciais e conforme aponta Sommerville (2011), por todos esses motivos, apresentam grande risco para serem substituídos.

Entre as principais funções disponíveis nos sistemas de atendimento estão: cadastro de ativos (moedas, funcionários, clientes), propostas e contratações de serviços (compra e venda de moedas ou ações, financiamentos, empréstimos, etc.), análise de risco, atualização de cotação de moedas, controle e restrição de compra e venda de títulos, etc.

O processo de manutenção de software da organização, possui o envolvimento de diversas áreas de TI, conforme citado no item 3.2 Áreas de Tecnologia da Informação da organização. As áreas atuam em conjunto, pois os ambientes são restritos a profissionais autorizados a realização de atividades pertinentes a sua área. O canal de comunicação para abertura e acompanhamento de chamados é o SD, e o cadastro de solicitações é realizado por profissionais da área de atendimento 1.

Os principais conceitos para o entendimento dos processos de manutenção de software da organização são:

## A. Classificação e categorização

Com o objetivo de organizar e gerar uma prioridade para cada chamado, as solicitações são classificadas em três diferentes tipos no SD: Incidente, Serviço e Problema. A utilização da classificação segue um escopo baseado no modelo de gerenciamento de serviços *Information Technology Infrastructure Library* (ITIL) (Chiari, 2016), onde a proposta é que a organização atinja os seus objetivos de negócios, utilizando apropriadamente os serviços de TI. Cada classificação, com exceção do Problema que herda as informações de Incidentes, contém um conjunto de categorias. As categorias são utilizadas para determinar a origem da solicitação e servem como um refinamento para os relatórios gerenciais.

Segue uma definição para cada classificação e as principais categorias associadas aos incidentes e serviços:

- Incidente: o software apresenta um *bug*, que impede o usuário de seguir com a sua atividade. A solução de contorno ou a análise para a solução definitiva é realizada através do incidente.  
Categoria: erro de escopo de projeto, problemas de infraestrutura, erro no sistema e erro na integração de aplicações.
- Serviço: são as solicitações não associadas a um *bug* no sistema.  
Categoria: extração de dados, esclarecimento de dúvidas, erro do usuário e falta de funcionalidade (melhoria).
- Problema: é a análise e a solução definitiva para um problema que gerou de um a vários incidentes. Está sempre associado a um ou mais incidentes, pois até que a solução definitiva seja aplicada, outros incidentes podem ocorrer. A abertura do problema é realizada pela equipe de manutenção de software.

A Tabela 7 apresenta as classificações, categorias e exemplos de ocorrências comuns em cada categoria. A correta categorização do chamado é fundamental para que o fluxo da solicitação seja conduzido conforme os riscos que ela apresenta para o negócio.

Tabela 7 - Exemplo de ocorrência por categoria

Classificação	Categoria	Exemplo de ocorrência
Incident	Escopo	Após uma mudança de um projeto o usuário não consegue mais realizar determinada operação.
	Infraestrutura	Um serviço de comunicação entre um sistema interno e um sistema externo, está inativo, gerando problemas no envio e recebimento de informações.
	Integração	Uma operação está liquidada, porém não consta no sistema que valida as liquidações realizadas.
Solicitação de serviço	Extração de dados	Usuário solicita extração de dados, com informações que não constam em relatórios disponíveis no sistema.
	Dúvida	O usuário não sabe como utilizar determinada funcionalidade do sistema e solicita ajuda.
	Erro do usuário	O usuário realizou um cadastro incorretamente e precisa alterar alguma informação da qual o sistema não permite.
	Falta de funcionalidade	Solicitação de nova tela, novo campo, nova regra

É possível associar a classificação proposta por Sommerville (2011), que divide as solicitações em três tipos distintos (Reparação de Falhas, Adequação ao Ambiente e Funcionalidade Adicional), com as classificações utilizadas pela organização estudada:

- Reparação de falhas: as classificações Incidentes e Problemas estão relacionadas às correções de erros de qualquer natureza;
- Adequação ao ambiente: a modificação de um software, para mantê-lo em funcionamento após mudanças externas, é realizada através da classificação Solicitação de Serviço;
- Funcionalidade adicional: as mudanças de requisitos nos softwares são classificadas como Solicitações de Serviço.

Para as solicitações de serviços associadas a categorias como extrações de dados e dúvidas, por exemplo, não há uma classificação para associação.

## **B. Tempo previsto para a conclusão do chamado**

Cada categoria possui um tempo previsto associado para a conclusão do chamado. O tempo serve para direcionar o profissional de manutenção, quanto às prioridades e riscos de cada tipo de solicitação.

## **C. Estado do chamado**

Muitas vezes o tempo previsto para a solução de um chamado, não é o suficiente para concluir-lo, devido a dependência que o analista possui com outras áreas. E por isso o sistema SD possui a opção de mudança do estado do chamado. Ao modificar o estado de um chamado, é necessário descrever o motivo da mudança, para que através da ferramenta, seja possível identificar qual a real situação da solicitação. As opções para a mudança de estado são:

- Enfileirado: é o estado o inicial da solicitação, onde ainda não é iniciada a contagem de tempo para a conclusão;
- Ativo: assim que um analista assume o chamado, o estado é modificado automaticamente para Ativo, e a contagem de tempo para a conclusão é iniciada;
- Pendente pelo fornecedor: aguardando retorno do fornecedor por motivos tais como, esclarecimento de dúvidas, implementação de solução;
- Pendente pelo usuário: aguardando retorno do usuário por motivos como esclarecimento de dúvidas, informações adicionais, testes;
- Aguardando implementação: a solicitação já foi realizada e validada pelo usuário no ambiente de homologação, e está aguardando a equipes de banco de dados ou / e infraestrutura, aplica-la no ambiente de produção;
- Encerrado: a solicitação foi atendida e o chamado foi concluído.

É importante que o analista modifique o estado de um chamado corretamente no momento que a mudança acontece, para que o chamado represente o tempo gasto

de atuação do analista, o tempo previsto não seja atingido indevidamente e os relatórios gerenciais apresentem resultados reais a qualquer momento.

#### **D. Abertura de chamados**

O fluxo de abertura de chamados para a área de manutenção de software é:

- O usuário (área de negócio) realiza uma solicitação via e-mail, para o atendimento 1 ou diretamente ao profissional da área de manutenção;
- Caso o usuário encaminhe a solicitação diretamente ao analista de manutenção, o analista deve encaminhar o e-mail para o atendimento 1, para que um chamado seja criado;
- Para a abertura do chamado a solicitação deve conter os seguintes dados: nome do usuário solicitante, nome do software de atendimento e uma descrição do problema ou serviço solicitado;
- O profissional do atendimento 1, analisa os dados e a necessidade de abrir o chamado. Caso falte algum dado, o profissional retorna à solicitação para o usuário. Caso a solicitação possua todos os dados e necessite de chamado o profissional cadastrá, classifica e categoriza o chamado no SD;
- O SD possui uma associação entre o software de atendimento e a equipe de manutenção responsável pelo sistema de atendimento;
- O chamado é automaticamente associado a fila de chamados da equipe responsável pelo sistema de atendimento;
- Um profissional de manutenção, assume o chamado pelo SD, e a partir desse momento o tempo para a conclusão passa a ser contabilizado;
- Se o chamado estiver categorizado como uma melhoria, o analista deve encerra-lo no SD, e cadastrá-lo no SM.

A Figura 4 representa o fluxo de abertura de chamados.

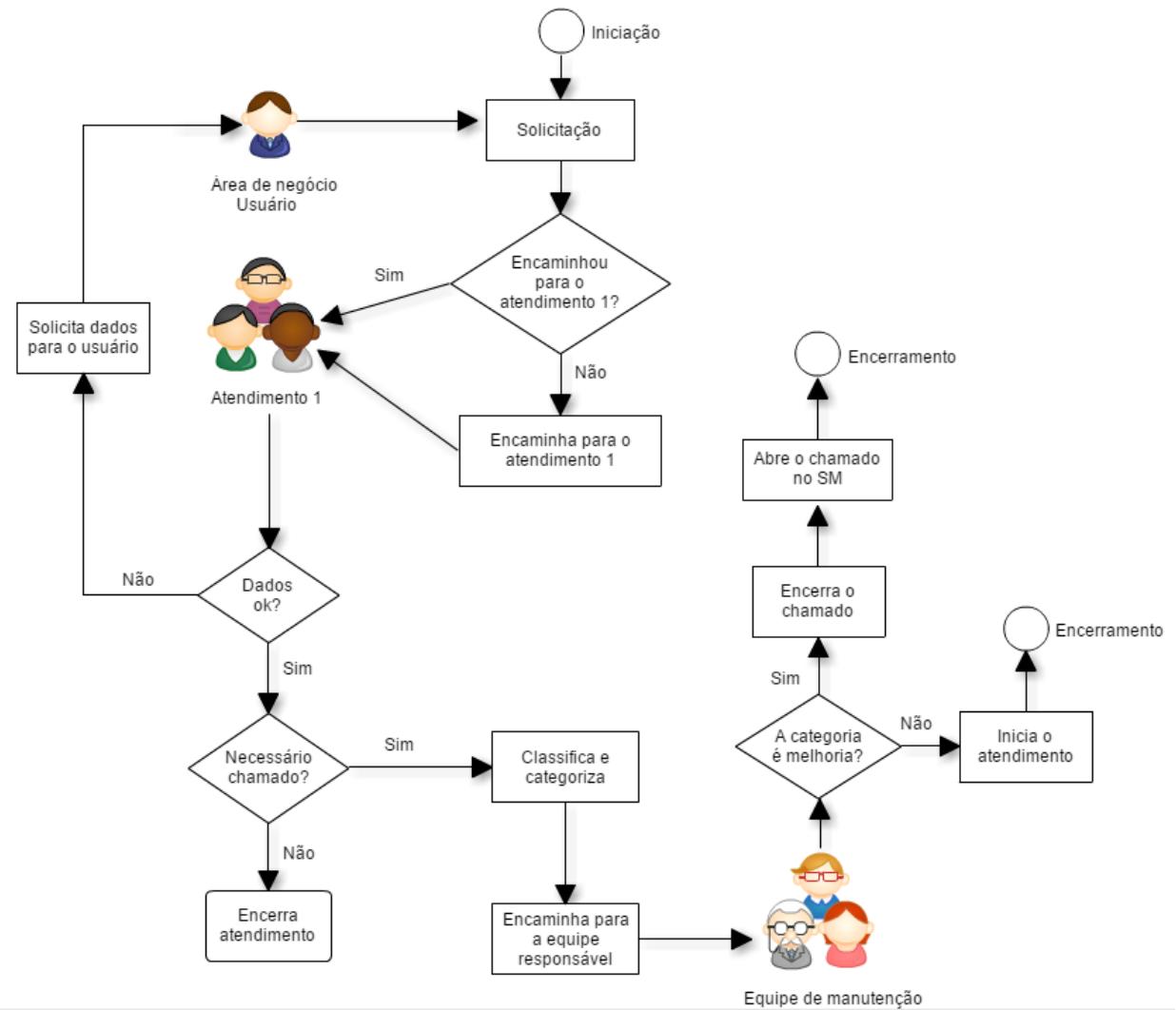


Figura 4 – Fluxo de abertura de chamados

A abertura do chamado impacta diretamente no tempo previsto de atendimento de cada solicitação, nas filas de *backlog*, prioridades, etc. Por isso a importância da correta classificação e categorização no SD. O registro incorreto de um chamado pode causar uma visão errônea da área de manutenção de software, prejudicando o controle e as tomadas de decisões da gestão.

## E. Fluxo de melhorias

As solicitações de melhoria seguem um fluxo diferente das outras solicitações. O fluxo foi definido pela gestão da área de manutenção de software da organização,

com o principal intuito de separar as demandas de melhorias da fila de *backlog*, tratando-as de forma distinta dos problemas, sem a associação de um tempo previsto de conclusão.

Inicialmente a melhoria é cadastrada no SD como solicitação de serviço e categoria melhoria. O cadastro da melhoria no SD, serve apenas para gerar um código de chamado para ela, e então o chamado é fechado no SD e cadastrado no sistema de acompanhamento de melhorias (SM). O SM possui uma função de priorização das demandas por área de negócio, dessa forma, todos os sistemas pertinentes a uma área de negócios, seguem uma única ordem de priorização de atendimento.

### 3.5 Atividades de manutenção de software

As atividades de manutenção de software da organização, se iniciam após a abertura do chamado e seguem os seguintes passos:

- Análise da solicitação: o analista analisa a demanda, e caso tenha dúvidas, entra em contato com o usuário solicitante para os esclarecimentos;
- Aplicação da solução: após o entendimento da solicitação o analista identifica a última versão da aplicação ou *script* de banco de dados disponível, verifica as alterações ou correções necessárias e aplica no ambiente de desenvolvimento;
- Alteração em homologação: o analista abre uma solicitação para a implementação das alterações no ambiente de homologação, e a solicitação é realizada por um profissional da área correspondente (banco de dados ou infraestrutura, por exemplo). Após a implementação o analista valida se as alterações foram realizadas no ambiente;
- Teste do usuário: com o sistema de atendimento modificado no ambiente de homologação, o analista encaminha para o usuário validar as alterações;
- Teste integrado: testes integrados com a participação de outras equipes, são realizados sempre que necessário, afim de garantir a integridade da aplicação e identificar possíveis impactos que a alteração pode causar em softwares integrados a aplicação modificada;

- Retorno do usuário: caso o usuário encontre algum problema durante os testes, a atividade retorna para o passo de análise da solicitação, até que a alteração esteja em conformidade com o que foi solicitado, e o usuário formalize que está de acordo por e-mail;
- Alteração em produção: o analista abre uma solicitação para a implementação das alterações no ambiente de produção, e a solicitação é realizada por um profissional da área correspondente (banco de dados ou infraestrutura, por exemplo). Após a implementação o analista verifica se o sistema está operacional e encaminha um e-mail para os usuários, verificarem as alterações no ambiente alterado;
- Acompanhamento: caso o usuário encontre algum problema na versão alterada em produção, o analista solicita o *downgrade* da alteração, que é realizado pelas equipes correspondentes (banco de dados ou infraestrutura, por exemplo).

A Figura 5, apresenta os passos das atividades de manutenção de software e a dependência que cada tarefa possui com outras áreas da organização.

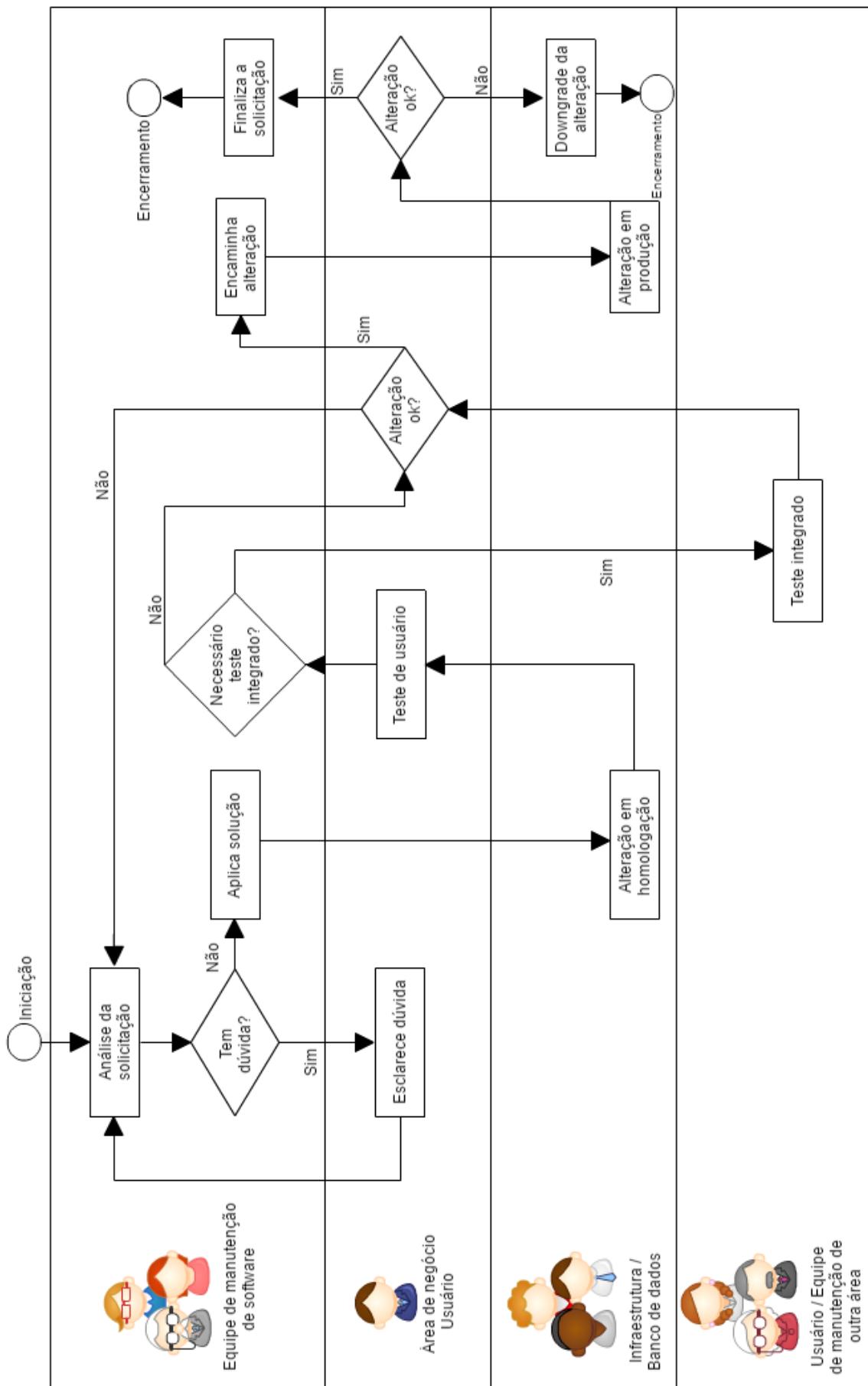


Figura 5 - Atividades de manutenção de software

### 3.6 Problemas na área de manutenção de software

A área de manutenção de software da organização possui um processo definido, porém ao analisar a execução das atividades do processo, assim como o comportamento dos profissionais e gestores, alguns fatores foram levantados como problemáticos, e passíveis de mudanças. É possível observar que em algumas equipes os processos funcionam de maneira mais efetiva que em outras.

Equipes estáveis são aquelas que conseguiram atuar em soluções definitivas de problemas que geravam muitos incidentes, e aos poucos as reincidências de chamados de erros diminuíram consideravelmente, dando espaço para uma maior atuação em chamados de melhorias. Essas equipes costumam responder rapidamente às solicitações dos usuários e conquistam a sua confiança, fator fundamental para um bom relacionamento entre as áreas. Por outro lado, as equipes instáveis encontram-se muitas vezes em meio ao caos, as filas de chamados de problemas são crescentes, as correções muito pontuais e as soluções definitivas pouco aplicadas, não há tempo para a atuação em melhorias. Os usuários estão normalmente insatisfeitos com a falta de retorno e de solução, e o relacionamento entre as áreas não é confortável.

Fazendo uma comparação de uma equipe estável e outra que se encontra em meio ao caos, pôde-se observar que a maneira com que o profissional encara e executa as atividades do processo, define os resultados alcançados em longo prazo.

Através de entrevistas com analistas, gestores e análises e relatórios da empresa, foi possível identificar alguns problemas encontrados na área de manutenção de software.

Para uma equipe instável as principais falhas encontradas com relação a execução das atividades do processo de manutenção de software foram:

- Atendimento sem chamado: o suporte ao usuário é realizado via telefone, e-mail ou mensagem, sem o encaminhamento da solicitação para a área de atendimento 1;
- Classificação / categoria incorreta: o chamado é cadastrado incorretamente e o analista não solicita a alteração;

- Falta de monitoramento: o estado dos chamados não é alterado corretamente; há chamados duplicados; chamados encerrados encontram-se como ativos;
- Demora no retorno ao usuário: o usuário não recebe retorno quanto as suas solicitações prontamente;
- Resistência aos processos: os profissionais se sentem na maior parte do tempo sobrecarregados de atividades e consideram que não há tempo para seguir os processos corretamente.

Os principais problemas levantados por analistas de manutenção de software, de equipes instáveis foram:

- Falta de apoio da gestão: a gestão não auxilia nas dificuldades e obstáculos diários;
- Falta de tempo para a realização de melhorias e soluções definitivas: as melhorias são tratadas com menor prioridade que os outros tipos de chamados, e os profissionais passam mais tempo corrigindo *bugs*, do que aplicando novas soluções;
- Falta de tempo para a utilização de processos: alguns profissionais argumentam não ter tempo suficiente para a execução adequada das atividades do processo.

De forma geral, analistas de diversas equipes, mencionaram alguns pontos comuns, que deveriam ser tratados:

- Requisitos insuficientes e constante mudança de requisitos: os usuários não sabem o que precisam realmente, e modificam constantemente a solicitação;
- Treinamento e desenvolvimento: os profissionais atuam com os mesmos sistemas e não se desenvolvem em outras tecnologias;
- Conhecimento sobre os sistemas de atendimento: os profissionais não têm conhecimento suficiente ou abrangente dos sistemas que atendem, e de suas estruturas e integrações;

- Treinamento para o usuário: em algumas áreas de negócio, a rotatividade de pessoas é frequente, e os usuários apresentam dificuldades para utilizar os sistemas;
- Falha na comunicação com a equipe de projetos: as equipes de projetos introduzem mudanças nos sistemas sem comunicar as equipes de manutenção de software e sem testes suficientes;

Com relação aos sistemas de atendimento, os profissionais identificaram como principal problema:

- Falta de documentação: grande parte dos sistemas não possui documentação ou está obsoleta.

A Tabela 8 apresenta os principais problemas de manutenção de software identificados na área, e as ocorrências que são os exemplos práticos.

Tabela 8 – Problemas na área de manutenção de software da organização

Problema	Ocorrência
Falha na execução do processo de manutenção de software	Atendimento sem chamado Cassificação / categorização incorretas Falta de monitoramento Demora no retorno ao usuário Resistência aos processos
Problemas encontrados por analistas de equipes instáveis	Falta de apoio da gestão Falta de tempo para a realização de melhorias e soluções definitivas Falta de tempo para a utilização de processos
Pontos levantados por equipes estáveis e instáveis (pontos em comum)	Requisitos insuficientes / mudança de requisitos Treinamento e desenvolvimento Preferência por atividades de desenvolvimento de software Conhecimento sobre os sistemas de atendimento Treinamento para o usuário
	Falha na comunicação
Problemas dos sistemas de atendimento	Falta de documentação

### 3.7 Planejamento de atividades para a implementação do processo Gerência de Projetos do MPS.BR nível G na organização

Reconhecendo as dificuldades da área a organização optou por melhorar os seus processos utilizando como base um modelo de melhoria formal de processos de software. O modelo MPS.BR se enquadrou nas necessidades da organização, por ser focado no mercado de software brasileiro e por conter mais níveis que o modelo CMMI, o que possibilita maior visibilidade quanto a evolução dos níveis de maturidade (MPS.BR, 2016).

O planejamento para implementação do processo Gerência de Projetos do MPS.BR nível G na organização, considerou o período de implementação do processo

(Abril a Dezembro de 2016), e as atividades para o cumprimento dos resultados esperados foram distribuídas dentro deste período, conforme a Tabela 9.

Tabela 9 – Cronograma de planejamento de atividades

Resultados Esperados	Atividades	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
GPR 1.	Definição do escopo									
GPR 2.	Análise de criticidade x tempo de atendimento dos chamados									
GRP 3.	Definição do modelo de atendimento									
GPR 4.	Análise de esforço e custo para a execução das tarefas									
GPR 5.	Definição de marcos e pontos de controle									
GPR 6. GPR 18.	Gerenciamento de riscos									
GPR 7.	Definição de papéis e responsabilidades									
GPR 8.	Levantamento e controle de recursos									
GPR 9.	Elaboração de documentos para treinamento									
GPR 9.	Definição de procedimentos									
GPR 9.	Implantação de ferramenta para controle de chamados									
GPR 10.	Definição do fluxo de atendimento									
GPR 11.	Reestruturação da equipe									
GPR 12. GPR 13. GPR 14. GRP 17. GPR 18.	Realização de reuniões recorrentes									
GPR 15.	Gerenciamento de riscos									
GPR 16.	Elaboração de matriz de papéis e responsabilidades									

As atividades representadas no cronograma pela cor verde, se referem a ações contínuas no processo de melhoria. E ao final da implementação do processo, os resultados esperados pelos atributos AP1.1 e AP2.1 devem ser atendidos, que

indicam se a execução do processo é gerenciada e se o processo produz os resultados definidos.

### **3.8 Considerações finais do capítulo**

A manutenção de software assume um papel cada vez mais importante dentro do ciclo de vida do software. A organização onde o estudo de caso foi realizado embora já tivesse um processo para as atividades de manutenção de software, identificou através de um diagnóstico de problemas da área, a necessidade de seguir um padrão de documentação de seus processos, além de padronizar os processos de manutenção de software.

Com base nas características do processo Gerência de Projetos do MPS-BR nível G, foi realizado um planejamento para alcançar os resultados do processo, contendo as atividades e suas referências periódicas estabelecidas.

## 4 ANÁLISE DOS RESULTADOS

### 4.1 Considerações iniciais

Nesse capítulo é detalhada a implementação do processo Gerência de Projetos do MPS.BR nível G – Parcialmente Gerenciado e os resultados obtidos.

A melhoria contínua iniciou em abril de 2016 e finalizou em dezembro de 2016. E contou com a participação dos seguintes papéis dentro da organização (Figura 6):

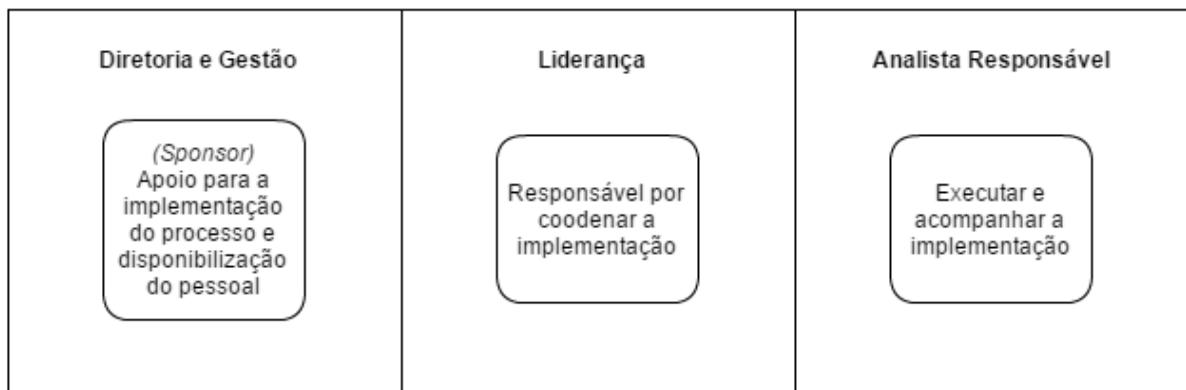


Figura 6 – Papéis e responsabilidades na implementação MPS.BR

### 4.2 Implementação e resultados do processo Gerência de Projetos

A implementação do processo Gerência de projetos do MPS.BR nível G – Parcialmente Gerenciado na organização, foi conduzida para solucionar os principais problemas encontrados na área e padronizar a forma com a qual as pessoas realizavam suas atividades.

Para a implementação de melhoria contínua nos processos de manutenção de software, um diagnóstico foi realizado na área para o entendimento do processo utilizado e identificação dos principais problemas, conforme detalhado no item 3.3 Área de manutenção de software da organização. As seguintes atividades foram realizadas considerando o cronograma de planejamento e as características e planejamento do processo Gerência de Projetos do MPS.BR nível G.

- GPR 1. O escopo do trabalho para o projeto é definido.

O escopo do projeto foi definido e documentado, contendo uma lista de objetivos específicos e atividades, servindo como um guia para atingir os propósitos e resultados do processo. Os principais objetivos contidos no escopo do projeto foram:

- Definição de papéis e responsabilidades;
- Definição de atividades;
- Padronização de atendimento aos chamados;
- Organização das filas de chamados;
- Monitoramento das filas de *backlog* dos chamados;
- Análise e tratamento de riscos;
- Acompanhamento das atividades através de relatórios e reuniões.

- GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados.

O tempo previsto de conclusão de chamados considera fatores como a criticidade, que tem relação com a necessidade do usuário, e a expectativa de conclusão do atendimento, de acordo com a necessidade. Quanto mais crítico o chamado, espera-se que mais rápido ele deva ser atendido e finalizado. A Tabela 10 apresenta exemplos dessa definição.

Tabela 10 – Tempo de conclusão dos chamados

Classificação	Categoria	Tempo de conclusão em horas
Incidente	Indisponibilidade do sistema	2
Incidente	Problema de desempenho	6
Incidente	Erro no sistema	6
Incidente	Erro de integração	24
Serviço	Esclarecimento de dúvidas	12
Serviço	Extração de dados	36
Serviço	Melhoria no sistema	168

- GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos. O modelo utilizado para o atendimento aos chamados foi definido como cascata, ou seja, as atividades do início ao fim do atendimento são realizadas de forma sequencial. A Figura 7 mostra os passos realizados.

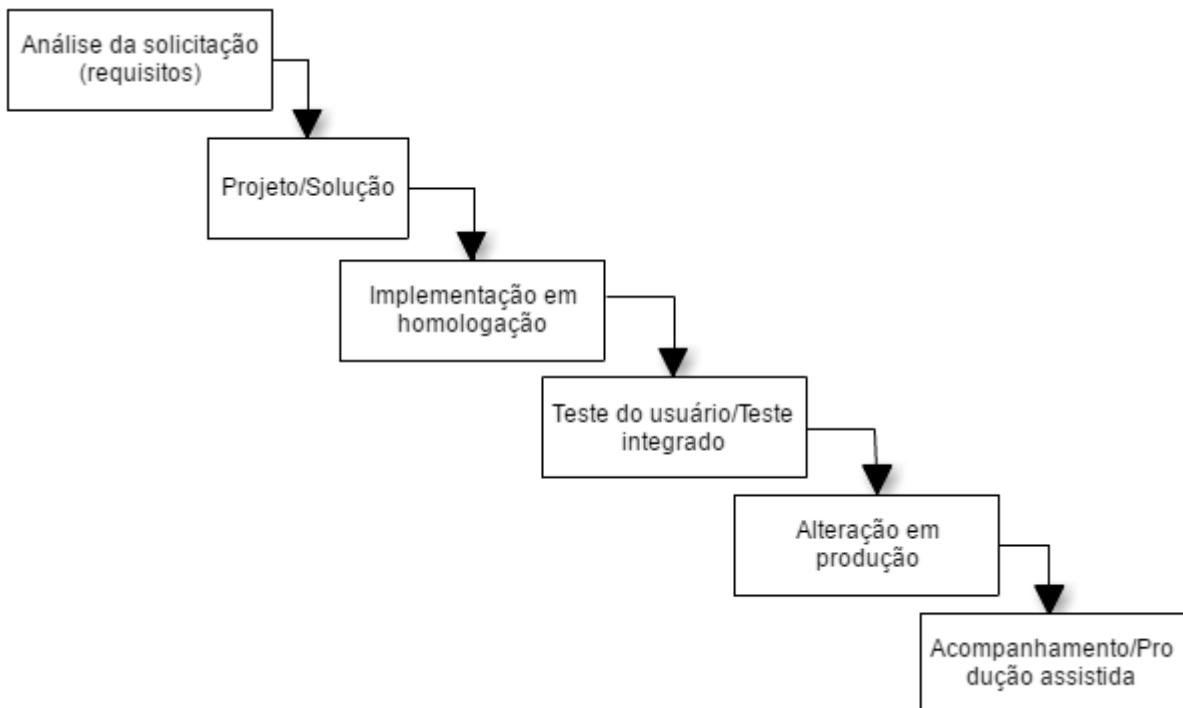


Figura 7 – Ciclo de manutenção de software na organização

- GPR 4. (Até o nível F) O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas.  
O custo é baseado na quantidade de horas trabalhadas por profissional. A área de negócio contrata a área de manutenção de software, considerando a quantidade de profissionais necessária para atender os chamados e janela de trabalho. Por exemplo, a área de câmbio contrata dois profissionais, pelo período de oito horas/dia em janelas de trabalho alternadas (horários diferenciados de atendimento).  
A quantidade de profissionais e horários de atendimento, é estabelecida conforme os dados históricos e conhecimento técnico dos profissionais.

- GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos.
  - Reuniões trimestrais ou sob demanda foram realizadas com a diretoria, para:
    - Alinhamento de TI com as estratégias de negócio;
    - Avaliação de oportunidades;
    - Avaliação do desempenho da área;
    - Gestão e monitoração do atendimento das expectativas;
    - Gerenciamento de crise e conflitos.
  - Reuniões mensais com a Gestão foram realizadas para:
    - Planejamento e acompanhamento de ações para implementação das estratégias definidas pela diretoria;
    - Avaliação do desempenho da equipe;
    - Identificação de oportunidades;
    - Proposição e acompanhamento de planos de ação, quando necessário;
    - Gerenciamento de crise e conflitos.
  - Reuniões quinzenais entre liderança, gestão e um responsável pela área de negócios foram realizadas para definições de ações de melhoria ou mudança e apresentação dos seguintes resultados:
    - Quantidade de chamados abertos nos últimos 15 dias;
    - Quantidade de chamados encerrados nos últimos 15 dias;
    - Tempo médio de atendimento.
  - Reuniões semanais entre os analistas e líderes foram realizadas para:
    - Revisão das filas de chamados, e auxílio pontual quanto ao esclarecimento ou direcionamento de dúvidas;

- Eliminação de impedimentos referente a atividades da responsabilidade de outras áreas (configuração de ambiente, disponibilização de acessos, entre outros).
- GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados.

O gerenciamento dos riscos era realizado pela liderança através de documentação em planilha, contendo as seguintes informações:

- Descrição do risco: detalhamento do risco;
- Gatilho: o que pode provocar o risco;
- Impacto (Alto, Médio, Baixo);
- Plano de contingência: como resolver o risco, caso ele ocorra.

Exemplo:

- Descrição do risco: indisponibilidade de profissional;
- Gatilho: ausência média ou pedido de demissão;
- Impacto: alto;
- Plano de contingência: treinar e capacitar recursos com alta disponibilidade.

- GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo.

Os papais e responsabilidades foram definidos de acordo com o nível de carreira, experiência e objetivo do profissional, e apresentados para a equipe conforme a figura 8.

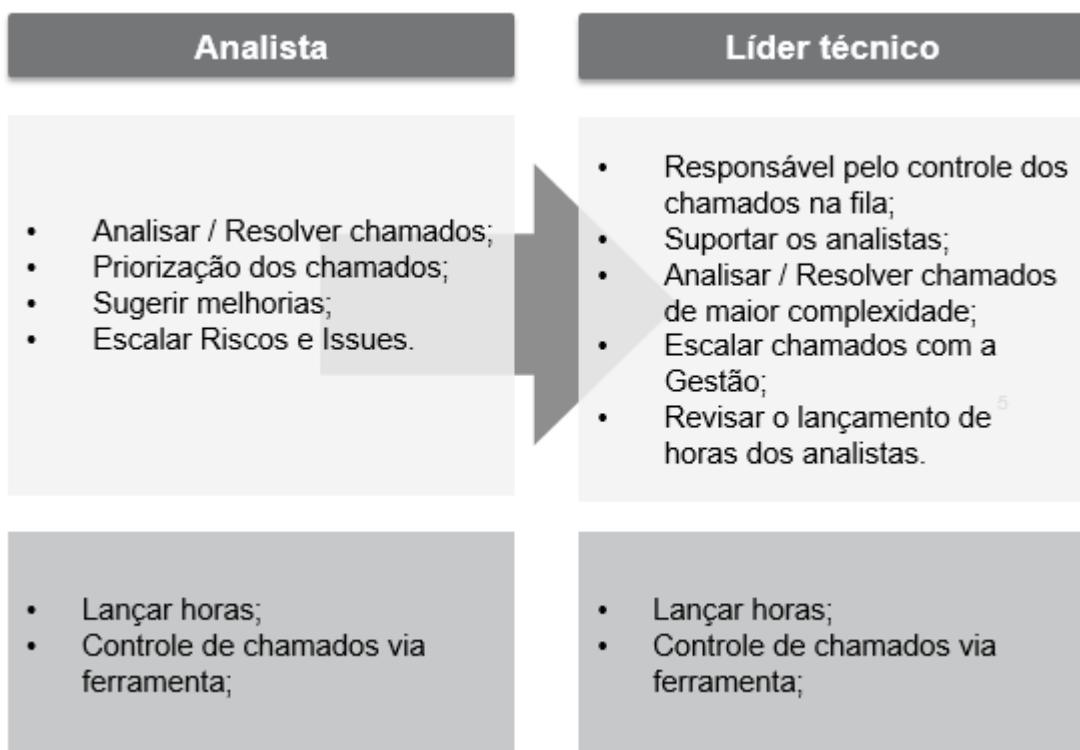


Figura 8 – Papéis e responsabilidades na organização

- GPR 8. (Até o nível F) Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados. Os recursos necessários para o desempenho das atividades, como por exemplo, máquinas, acessos, softwares e configurações necessárias para o atendimento aos chamados, são disponibilizados antes do início das atividades.
- GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
  - Através de uma ferramenta colaborativa online do projeto, todo o material gerado, como documentação de treinamentos e procedimentos foi armazenado e disponibilizado para a equipe.

- Através da ferramenta de chamados, foram disponibilizados relatórios *ad hoc* (para um fim específico), onde cada recurso tem visibilidade somente às informações pertinentes ao seu perfil. A consulta ao relatório era realizada através de filtros, como:
  - Classificação do chamado (incidente, serviço ou problema);
  - Data de abertura ou encerramento;
  - Usuário solicitante;
  - Área de negócio;
  - Sistema.
- GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos.  
Foi criado um material com um detalhamento do fluxo de atendimento e compartilhado com todas as áreas envolvidas, para que todas elas tenham um critério estabelecido para o atendimento dos chamados. O documento seguiu o modelo do item 3.5 Atividades de manutenção de software.
- GPR 11. A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados.  
Uma reestruturação foi realizada na equipe adequando o perfil dos profissionais às atividades desempenhadas, conforme os detalhes da figura 8 – Papéis e responsabilidades na organização.
- GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido.  
As reuniões planejadas foram realizadas conforme detalhamento da atividade GPR 5, respeitando a frequência estabelecida.
- GPR 13. O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado.

As reuniões para monitoramento do projeto foram realizadas conforme o detalhamento da atividade GPR 5, respeitando a frequência estabelecida.

- GPR 14. Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado.  
A liderança realizou o monitoramento das necessidades do projeto através das reuniões semanais, quinzenais e mensais, detalhadas na atividade GPR 5.
- GPR 15. Os riscos são monitorados em relação ao planejado.  
A liderança monitorou os riscos, conforme a planilha adotada para esse controle. Detalhamento na atividade GPR 6.
- GPR 16. O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido.  
Foi elaborado um documento com a matriz de papéis e responsabilidades das partes interessadas e disponibilizado na ferramenta de colaboração *online* da organização. Alterações foram realizadas no documento, quando houve mudança na estrutura organizacional, permitindo que todos os envolvidos tivessem acesso a informação atualizada.
- GPR 17. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento.  
As reuniões estabelecidas para o acompanhamento das atividades foram realizadas conforme detalhados na atividade GPR 5.
- GPR 18. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas.

Através das reuniões semanais entre liderança e analistas, foi possível mapear problemas que a equipe enfrentava, relacionado a atuação com outras áreas, como por exemplo:

- Demora para a liberação de acesso;
- Demora para a instalação de softwares necessários para o atendimento de um chamado.
- GPR 19. Ações para corrigir desvios em relação ao planejado e para prevenir repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.

Um mecanismo foi criado para que chamados que estivessem próximos de atingir o tempo previsto para conclusão, fossem encaminhados para o e-mail do líder da equipe, em forma de alerta. Essa medida foi importante para:

- Controle da liderança quanto aos impedimentos encontrados;
- Atuação imediata para que o tempo previsto fosse respeitado;
- Identificação de mudanças (criticidade x tempo de conclusão);
- Identificação de analistas que não encerravam os chamados na ferramenta após a finalização do atendimento.

Os resultados obtidos com a aplicação do processo Gerência de Projetos, conforme os atributos AP1.1 O processo é executado e AP2.1 A execução do processo é gerenciada, foram:

- O processo produz os resultados definidos.  
Os resultados definidos no escopo do projeto foram atingidos.
- Existe uma política organizacional estabelecida e mantida para o processo.  
Foram elaborados documentos servindo como um guia para atingir os propósitos e resultados esperados com a realização das atividades:
  - Guia de atendimento;

- Matriz organizacional;
- Papéis e responsabilidades.

Os documentos foram disponibilizados para a área, através de uma plataforma de colaboração *online*, dessa forma os colaboradores passaram a seguir os procedimentos adotados e compartilhados.

- A execução do processo é planejada (O planejamento deve incluir identificação e disponibilização dos recursos e informações necessárias para a execução do processo, definição, atribuição e comunicação das responsabilidades pela execução do processo e planejamento da comunicação entre as partes interessadas).
  - Foi realizada uma apresentação para a equipe dos papéis e responsabilidades de cada recurso e dos processos a serem seguidos;
  - Os analistas foram orientados a comunicar à liderança qualquer dificuldade de interação com outras áreas.
- A execução do processo é monitorada em relação ao planejado e, quando necessário, ajustes são realizados.

A execução do processo foi monitorada através de reuniões.
- As pessoas que executam o processo estão preparadas para executar suas responsabilidades.

As pessoas receberam treinamentos, documentos, orientação e acompanhamento para a realização de suas atividades. Os treinamentos e orientações foram:

  - Guia de atendimento;
  - Papéis e responsabilidades;
  - Atendimento às solicitações do usuário, somente através da ferramenta de chamados (formalização);
  - Monitoramento de chamados através de relatórios *ad hoc*.

- As atividades, o status e os resultados do processo são revistos com a gerência de nível superior e são tratadas questões críticas.

O monitoramento das atividades foi realizado através de reuniões trimestrais ou sob demanda com a diretoria e mensais com a gestão. Nas reuniões foram apresentados assuntos como:

- Quantidade de chamados abertos e fechados;
- Evolução no atendimento dos chamados;
- Eficiência no atendimento por funcionário (quantidade de chamados atendidos e tempo de atendimento).

O tratamento de questões críticas foram realizados sob demanda e monitorados pela planilha de risco.

#### **4.3 Considerações finais do capítulo**

A organização atingiu os resultados esperados conforme o cronograma de planejamento das atividades e características do processo Gerência de Projetos do MR.MPS nível G, como também a melhoria no gerenciamento do atendimento aos chamados. A documentação dos processos que são executados pela área de manutenção de software foi criada e compartilhada com os profissionais, o que facilitou na execução das atividades e na disseminação de conhecimento para novos colaboradores. As reuniões para monitoramento, revisões e análises de pontos críticos foram importantes para a área conseguir atingir os objetivos do escopo do projeto. A principal dificuldade considerada para a implementação do processo foi a resistência das pessoas às mudanças, pois alguns profissionais se mostraram inicialmente resistentes a aderir os processos, o que foi contornado com a contribuição da liderança em acompanhar e orientar os profissionais.

## 5 CONSIDERAÇÕES FINAIS

### 5.1 Contribuições do trabalho

O trabalho apresenta a aplicação de um modelo de melhoria, incluindo a apresentação detalhada das atividades da área de manutenção de software de uma organização do mercado de investimentos, o fluxo realizado utilizado para o atendimento das solicitações, os problemas enfrentados na área, a partir do histórico de projetos e lições aprendidas e a identificação dos principais passos para a melhoria dos processos.

Para a implantação de um dos níveis para melhoria contínua de processos na organização, o comprometimento das pessoas foi essencial, pois elas responsáveis por planejar, apoiar, executar e manter um cronograma de atividades, assim como gerenciar e atuar em questões críticas, adequando o ambiente organizacional a um novo modelo de processos, a fim de melhor atender as necessidades e expectativas do usuário.

Os processos organizacionais devem passar por uma análise e planejamento para alcançar os resultados esperados de um modelo de melhoria de processos e a melhoria nos processos de software quando aplicada, não atinge somente a área em questão, que no caso do estudo de caso se tratou da área de manutenção de software, mas o setor de TI e negócios como um todo.

### 5.2 Trabalhos futuros

Para trabalhos futuros podem ser avaliadas para projetos de manutenção de software, aplicações de modelos voltados ao gerenciamento serviços como por exemplo a metodologia ITIL, que possui o *framework* para gerenciamento de serviços de TI mais utilizado mundialmente (Chiari, 2016) e os modelos de referência para melhoria de serviços: Modelo do Processo de Serviços (MPS-SV) e o *Capability Maturity Model Integration for Services* (CMMI-SVC), que consideram que a melhoria

das práticas de serviços são chaves para um melhor desempenho, aumento da satisfação do cliente e da lucratividade do setor (MPS.BR, 2016).

Outro trabalho futuro a ser considerado, é a realização da avaliação da implementação do processo Gerência de Projetos do MPS.BR nível G, na organização do estudo de caso descrito nesse trabalho, para a confirmação do processo implementado de acordo com o Método de Avaliação MPS (MA-MPS) detalhado no Guia de Avaliação da Softex (MPS.BR, 2015). E também a avaliação de possíveis adaptações aos modelos de melhoria contínua de software, considerando projetos de manutenção de software.

## REFERÊNCIAS

- BASUMATARY, B.; BARUAH, N.; BORGOHAIN, H. **A Study of Software Process Improvement Approaches Proposed for SMEs.** India: International Journal of Emerging Trends Technology in Computer Science, 2014. 3 v.
- BERTOLINI, C.; SILVEIRA, S.; CERATTI, P. **Implementação do MPS.BR na empresa Digifred Sistemas para Gestão Pública: Um Estudo de Caso.** 2016. Disponível em: <<http://revistas.setrem.com.br/index.php/reabtic/article/view/149>>. Acesso em: 12 Abr. 2017.
- BOEHM, B.; BROWN, J.; LIPOW, M. **Quantitative evaluation of software quality.** EUA: IEEE Computer Society Press Los Alamitos, 1976.
- CHIARI, R. **O QUE É ITIL.** 2016. Disponível em: <<http://www.mundoitil.com.br/>>. Acesso em: 24 Nov. 2016.
- CURTIS, B.; HEFLEY, B.; MILLER, S. **People Capability Maturity Model.** EUA: Software Engineering Institute, 2009.
- GLASS, R. **Frequently Forgotten Fundamental Facts about Software Engineering.** China: IEEE Computer Society, 2001. p. 112 – 111.
- FUGGETA, A. **Software Process: A Roadmap.** Italy, 2000.
- HARON, A.; SAHIBUDDIN, S.; HARUN, M.; ZAKARIA, N.; MAHRIN, M. **The Important Role of People, Process and Technology during Software Project Requirement.** Singapura: International Journal of Machine Learning and Computing, 2013. 3 v.
- HUFF, N.; GLOVER, C.; HILLMAN, M. **How software maintenance fees are siphoning away your IT budget—and How to Stop It.** 2014. Disponível em: <[https://www.accenture.com/t20150523T024819\\_\\_w\\_\\_/us-en/\\_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Dualpub\\_2/Accenture-How-Software-Maintenance-Fees-are-Siphoning-IT-Budget-Procurement-BPO.pdf](https://www.accenture.com/t20150523T024819__w__/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Dualpub_2/Accenture-How-Software-Maintenance-Fees-are-Siphoning-IT-Budget-Procurement-BPO.pdf)>. Acesso em: 11 Set. 2016.

CROLL, P.; PIGOSKI, T.; MOORE, J. **INTERNATIONAL STANDARD ISO/IEC 14764: IEEE Std 14764-2006. Software Engineering — Software Life Cycle Processes — Maintenance.** CH: IEEE-SA Standards Board, 2006.

KUHRMANN, M.; BEECHAM, S. **Artifact-Based Software Process Improvement and Management: A Method Proposal:** International Conference on Software and System Process. EUA: ACM, 2014. p. 119 – 123.

KUHRMANN, M.; DIEBOLD, P.; MUNCH, J. **Software process improvement: A systematic mapping study on the state of the.** EUA: PeerJ Computer Science, 2016.

LEWIS, W. **Software Testing and Continuous Quality Improvement.** 2. ed. EUA: CRC Press, 2004.

LYNCH, J. **Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch.** 2015. Disponível em: <<https://www.infoq.com/articles/standish-chaos-2015>>. Acesso em: 19 jul. 2016.

LAPORTE, C. Y.; O'CONNOR, R. V. **Software Process Improvement in Graduate Software Engineering Programs:** 1st International Workshop on Software Process Education, Training and Professionalism. Suécia, 2015.

**MPS.BR - Melhoria de Processo do Software Brasileiro:** Guia de Avaliação Parte II: Requisitos para Instituições Avaliadoras e Avaliadores MA-MPS. Brasil: Softex, 2015.

**MPS.BR - Melhoria de Processo do Software Brasileiro:** Guia Geral MPS de Software. Brasil: Softex, 2016.

MATTURO, G.; SAAVEDRA, M. G. **Considering People CMM for Managing Factors that Affect Software Process Improvement:** IEEE Latin America Transaction. Uruguai: IEEE Journals & Magazines, 2012. 1603 p. 10 v.

MENS, T.; DEMEYER, S. **Software Evolution.** EUA: Springer-Verlag Berlin Heidelberg, 2008.

MOHAN, P.; SHANKAR, A.; JAYASRIDEVI, K. **Quality Flaws: Issues and Challenges in Software Development.** India: International Institute for Science, Technology and Education (IISTE), 2012, vol. 3, n.12.

PAVEL, C.; MCELRAVEY, J. **Globalization in the financial services industry**. Chicago: Federal Reserve Bank of Chicago, 1990.

PRESSMAN, R. **Software Process Impediment**: A Practitioner's Approach. Orange: IEEE Journals & Magazines, 1996. p. 16-17. 13 v.

SILVEIRA, V.; GUIMARÃES, L.; ABRAÃO, H. **Os Modelos de Maturidade e a Gestão de Pessoas: O Modelo P-CMM**: XXXI Encontro da ANPAD. Brasil: ANPAD, 2007.

SNEED, M. **Offering software maintenance as an offshore service**. In: 24th IEEE International Conference on Software Maintenance (ICSM). Beijing, China: IEEE Computer Society, 2008.

SOFTWARE ENGINEERING INSTITUTE. **CMMI® for Development, Version 1.3: Improving processes for developing better products and services**. EUA: Carnegie Mellon University, 2010.

SOMERVILLE, I. **Software Engineering**. 9. ed. EUA: Pearson Education, 2011.

SRIVASTAVA, S.; AWASTI, A. **Software Process Improvement**. India: VSRD International Journal of Computer Science & Information Technology, 2014. p. 156 – 158, 4 v.

ZETLIN, B. **How to balance maintenance and IT innovation**. 2013. Disponível em: <<http://www.computerworld.com/article/2486278/it-management/how-to-balance-maintenance-and-it-innovation.html>>. Acesso em: 19 Set. 2016.