

FELIPE TAVARES JACOBUCCI

**UMA PROPOSTA PARA MELHORIA DO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE EM PEQUENAS EMPRESAS**

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
2015

FELIPE TAVARES JACOBUCCI

**UMA PROPOSTA PARA MELHORIA DO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE EM PEQUENAS EMPRESAS**

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: Prof. MSc. Sarah Kohan

São Paulo
2015

FICHA CATALOGRÁFICA

DEDICATÓRIA

*Dedico este trabalho a minha noiva
Bruna, pelo carinho e suporte, e a
meus pais, Lourival e Solange, pelo
exemplo de caráter e dedicação*

AGRADECIMENTOS

Agradeço ao PECE – Programa de Educação Continuada em Engenharia pela qualidade de ensino e por proporcionar esta importante oportunidade de crescimento acadêmico.

Meus sinceros agradecimentos à minha orientadora Professora MSc Sarah Kohan, que me auxiliou durante este trabalho, além de me proporcionar uma série de ensinamentos.

Agradeço a empresa Brasil Risk, a sua cúpula diretora que permitiu a realização deste trabalho, e em especial ao meu gestor Sylvio Périgo pela boa vontade, compreensão e apoio demonstrado durante confecção deste trabalho.

RESUMO

As pequenas empresas desenvolvedoras de software buscam cada vez mais melhorar sua competitividade através do reconhecimento como produtoras de software com qualidade. Devido à grande concorrência neste setor, um dos principais diferenciais competitivos é a adoção, avaliação e certificação de um processo de software baseado em boas práticas da engenharia de software. Porém, devido a restrições financeiras e estruturais, as pequenas empresas encontram dificuldades para atingir este objetivo. Este trabalho propõe um processo de software para pequenas empresas. Para isto, foram comparados três padrões de processo de software, sendo eles o CMMI-DEV, MPS.BR-SW e ISO/IEC 29110. O padrão mais adequado foi escolhido como referência e implementado em uma pequena empresa. O processo foi aplicado em um projeto e os resultados deste estudo de caso foram analisados. Com isto, o processo proposto neste trabalho pode servir como base para pequenas empresas que desejam melhorar seu processo de software.

Palavras-chave: Engenharia de Software, Melhoria do Processo, Processo de Software, Padrões de Processos

ABSTRACT

Small software development enterprises are increasingly seeking to improve their competitiveness by distinguishing themselves as producers of high quality software. Heavy competition in this field drives these companies to seek for a competitive advantage through the adoption, evaluation and certification of better software engineering process based on industry best practices. This work proposes one such software process for small enterprises. To meet this objective, three process standards, CMMI-DEV, MPS.BR-SW and ISO/IEC 29110, were compared and the most appropriate standard was chosen as a process reference and implemented in a small enterprise. Then this process was applied on a single real world project and the results of this case study were analyzed. Based on these results, the process shows promise as a basis for other small enterprises who wish to improve their software processes.

Keywords: Software Engineering, Process Improvement, Software Process, Process Standards

LISTA DE ILUSTRAÇÕES

Figura 1 - Camadas da Engenharia de software.....	17
Figura 2 - Componentes do modelo MPS.....	28
Figura 3 - Estrutura da norma ISO/IEC 29110	35
Figura 4 - O processo da norma ISO/IEC 29110	39
Figura 5 - Atividades do processo de gerenciamento de projetos (PM)	42
Figura 6 - Atividades do processo de implementação de software (SI)	46
Figura 7 - Organograma da área de TI.....	54
Figura 8 - Organograma do departamento de desenvolvimento de software....	55
Figura 9 - O atual processo de desenvolvimento	57
Figura 10 - Fraquezas sob a ótica da equipe	59
Figura 11 - Fraquezas, sob a ótica da diretoria	60
Figura 12 - Status dos projetos.....	61
Figura 13 - Projetos atrasados	61
Figura 14 - Metodologia de implantação do processo PM.....	65
Figura 15 - Fluxo de atividades do novo processo	69
Figura 16 - Estrutura do repositório do projeto.....	72
Figura 17 - Fluxo de atividades da solicitação de mudança.....	81
Figura 18 - Matriz de rastreabilidade dos requisitos	82

LISTA DE TABELAS

Tabela 1 - Níveis de maturidade do CMMI-DEV	25
Tabela 2 - Áreas de processo do CMMI-DEV	26
Tabela 3 - Níveis de maturidade MPS	29
Tabela 4 - Atributos de processo	30
Tabela 5 - Níveis de maturidade MPS.BR-SW	31
Tabela 6 - ISO/IEC 29110 por público alvo	34
Tabela 7 - Relação de Grupos de Perfis Genéricos.....	38
Tabela 8 - Relação de papéis.....	41
Tabela 9 - Atividades e objetivos do processo PM	44
Tabela 10 - Produtos de trabalho do processo PM	45
Tabela 11 - Atividades e objetivos do processo SI	48
Tabela 12 - Produtos de trabalho do processo SI	49
Tabela 13 - Matriz de papéis e responsabilidades.....	56
Tabela 14 - Principais falhas do processo de desenvolvimento atual	62
Tabela 15 - Atual situação em relação aos objetivos do processo PM	66
Tabela 16 - Atual situação em relação às atividades do processo PM.....	67
Tabela 17 - Atual situação em relação aos produtos de trabalho do processo PM	67
Tabela 18 - Produtos de trabalho.....	70
Tabela 19 - Resultados da pesquisa de satisfação	87

LISTA DE ABREVIATURAS E SIGLAS

VSE	<i>Very small entities</i>
CMMI	<i>Capability Maturity Model Integration</i>
SEI	<i>Software Engineering Institute</i>
MPS.BR	Melhoria do Processo de Software Brasileiro
ISO	<i>International Organization for Standardization</i>
IEC	<i>International Electrotechnical Commission</i>
ABNT	Associação Brasileira de Normas Técnicas
SVN	<i>Apache Subversion</i>

SUMÁRIO

1.	INTRODUÇÃO	13
1.1	Motivações	13
1.2	Objetivo	14
1.3	Justificativas	14
1.4	Estrutura do Trabalho	15
2.	REVISÃO BIBLIOGRÁFICA	16
2.1	Engenharia de Software	16
2.2	Processo de Software e Padrões de Processos	18
2.3	Gerenciamento de Projetos	19
2.4	Requisitos de Software	20
2.5	Verificação e Validação	22
2.6	Solicitações de Mudança	22
2.7	Considerações do Capítulo	23
3.	BOAS PRÁTICAS PARA DESENVOLVIMENTO DE SOFTWARE	24
3.1	CMMI-DEV	24
3.2	MPS.BR-SW.....	27
3.3	ABNT NBR ISO/IEC 29110-4.....	32
3.3.1	Processo de Gerenciamento de Projetos (PM)	42
3.3.2	Processo de Implementação de Software (SI)	46
3.4	Considerações do Capítulo	50
4.	ESTUDO DE CASO: APLICAÇÃO DAS BOAS PRÁTICAS EM UMA PEQUENA EMPRESA.....	53
4.1	A Empresa.....	53
4.2	Situação Atual.....	58
4.3	Metodologia para Melhoria do Processo.....	63

4.4	O Novo Processo de Desenvolvimento	69
4.4.1	Configurar o Repositório do Projeto.....	71
4.4.2	Criar a Declaração de Trabalho	73
4.4.3	Definir Requisitos de Software.....	75
4.4.4	Criar Plano do Projeto.....	76
4.4.5	Aprovação do Plano do Projeto	77
4.4.6	Registrar Status do Projeto	78
4.4.7	Atualizar o Progresso do Projeto Junto ao Cliente	79
4.4.8	Registrar e Acompanhar Mudanças.....	80
4.4.9	Validação	81
4.4.10	Encerrar o Projeto	83
4.5	Considerações do Capítulo	83
5.	CONSIDERAÇÕES FINAIS	84
5.1	Contribuições do Trabalho	84
5.2	Trabalhos Futuros	88
	REFERÊNCIAS.....	89
	GLOSSÁRIO.....	91

1. INTRODUÇÃO

Este capítulo apresenta as motivações, objetivos, justificativas e a estrutura do trabalho.

1.1 Motivações

O termo VSE (*very small entity*) é usado para qualificar as pequenas organizações produtoras de bens e/ou serviços. O papel desempenhado pelas VSEs é fundamental na economia de todos os países, uma vez que elas fornecem serviços e produtos para uma grande variedade de nichos de mercado (ABNT NBR ISO/IEC 29110-4, 2012).

De acordo com estudo publicado pelo SEBRAE (2011), existem 8,9 milhões de VSEs no Brasil, que geram 27% do PIB e são responsáveis por 52% dos empregos formais. E segundo relatório da ABES (2014), há cerca de 11.230 empresas na área de tecnologia da informação no país, sendo que 93% delas são VSEs.

Falando especificamente das VSEs que produzem software, muitas vezes seus componentes são usados ou integrados por produtos de organizações maiores. Desta forma, a entrega de um produto de baixa qualidade ou fora do prazo pode acarretar grandes prejuízos, tanto para a VSE quanto para seus clientes (LAPORTE, 2013).

Tal risco está relacionado às VSEs devido à baixa adoção de padrões e melhores práticas de desenvolvimento e controle da qualidade. O trabalho em uma VSE geralmente é feito de maneira informal e com pouca documentação. Muitas VSEs trabalham com orçamento restrito e quadro de funcionários reduzido, tornando a adoção de um padrão de desenvolvimento de software muito custoso e complexo.

De acordo com o SEI (*Software Engineering Institute*), órgão ligado à Universidade *Carnegie Mellon*, “a qualidade de um sistema ou produto é altamente influenciado pela qualidade do processo utilizado para o seu desenvolvimento e manutenção” (CHRISSIS et al, 2003). Com base nesta afirmação, quanto melhor o processo de desenvolvimento de software de uma empresa, melhor serão seus resultados e melhor será sua competitividade no mercado.

Como forma de melhorar o processo de software, muitas empresas buscam modelos de processo para desenvolver software com qualidade. Os modelos de referência trazem boas práticas reconhecidas no mercado, podendo ser usados para guiar o processo de desenvolvimento seguindo as práticas descritas, e também com a finalidade de se obter uma certificação, ou seja, um selo de qualidade aceito por todo mercado.

1.2 Objetivo

O objetivo deste trabalho é demonstrar como pequenas organizações produtoras de software podem melhorar a qualidade do seu processo de desenvolvimento, e por consequência, a qualidade de seu software, através da adoção de boas práticas de engenharia de software.

Para cumprir este objetivo, este trabalho propõe um o processo de desenvolvimento de software para pequenas empresas, e a metodologia utilizada durante a sua implantação. Para isto, alguns modelos de referência serão comparados sob o ponto de vista da adoção por pequenas empresas. Em seguida um estudo de caso será feito com base no modelo mais adequado. Finalmente, os resultados serão analisados a fim de se identificar pontos positivos, caso existam.

1.3 Justificativas

De acordo com Pressman (2009), um processo de software é uma coleção de atividades, ações e tarefas que devem ser executadas para que o software seja entregue em tempo hábil e com qualidade suficiente para satisfazer os *stakeholders*. Portanto uma empresa que possua um processo bem definido tem maior chance de entregar software de qualidade e aumentar sua competitividade.

Entre os benefícios em se adotar um processo de desenvolvimento baseado em padrões de processos, podemos relacionar o ganho de qualidade proporcionado pelas boas práticas de engenharia de software, melhorias na organização e na comunicação interna da equipe, o ganho de competitividade fruto da avaliação e certificação do processo.

Portanto, pretende-se que este trabalho possa relacionar as boas práticas da engenharia de software à realidade das VSEs, além de apoiar as pequenas empresas na busca por uma maior competitividade e maior qualidade de seus produtos de software.

1.4 Estrutura do Trabalho

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

O Capítulo 2 REVISÃO BIBLIOGRÁFICA apresenta uma revisão dos conceitos, definições e estudos relevantes ao tema abordado neste trabalho.

O Capítulo 3 BOAS PRÁTICAS PARA O DESENVOLVIMENTO DE SOFTWARE apresenta e descreve os modelos de referência CMMI-DEV, MPS.BR-SW e ISO/IEC 29110-4.

O Capítulo 4 ESTUDO DE CASO: APLICAÇÃO DAS BOAS PRÁTICAS EM UMA PEQUENA EMPRESA representa o desenvolvimento do trabalho, onde o modelo de referência selecionado é aplicado a uma VSE.

O Capítulo 5 CONSIDERAÇÕES FINAIS conclui o trabalho. É dividido em duas partes, onde a primeira descreve os resultados do estudo de caso, e a segunda parte relaciona algumas propostas para futuros trabalhos.

REFERÊNCIAS relaciona os trabalhos, artigos, livros e demais referências usadas neste trabalho.

GLOSSÁRIO relaciona os termos utilizados neste trabalho.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta uma revisão dos conceitos, definições e estudos relevantes ao tema abordado neste trabalho.

2.1 Engenharia de Software

O termo Engenharia de Software possui diversas definições, porém de forma geral podemos afirmar que se trata do estudo e aplicação da engenharia no design, construção e manutenção de software. A engenharia de software define as boas práticas empregadas na produção de produtos de software.

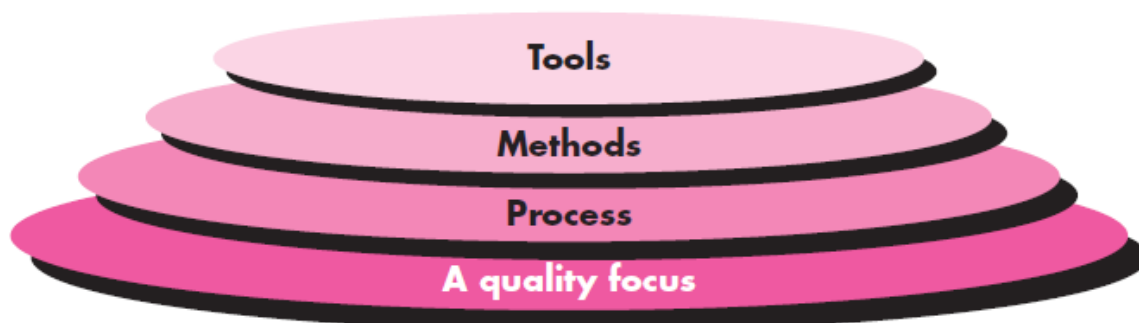
Fritz Bauer definiu Engenharia de Software, na Conferência *NATO Science Committee* em 1969, como “criação e a utilização de sólidos princípios de engenharia a fim de se obter softwares econômicos, que sejam confiáveis e que trabalhem de forma eficiente em máquinas reais”.

O Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), maior organização mundial de profissionais técnicos do mundo, define a engenharia de software da seguinte forma:

“A aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento e manutenção de software.” (IEEE, 1980)

De acordo com Pressman (2009), a engenharia de software é uma tecnologia dividida em camadas, como demonstrado na figura 1.

Figura 1 - Camadas da Engenharia de software



Fonte: Software Engineering – A Practioner's Approach (2009)

Ainda segundo o autor, a qualidade é o principal objetivo da engenharia de software. A camada de processos (*Process*) unifica as camadas com maior enfoque técnico, e permite o desenvolvimento de software de forma racional e em tempo hábil. A camada de Métodos (*Methods*) fornece o passo a passo técnico necessário a construção de softwares, e a camada de Ferramentas (*Tools*) fornece suporte de automação ou semi-automação para os processos e métodos.

Pressman (2009) define 5 atividades que servem de base para um processo de engenharia de software, que são aplicáveis a qualquer projeto, independentemente de seu tamanho ou complexidade. Estas 5 atividades são:

- Comunicação: o objetivo desta atividade é comunicar e colaborar com o cliente e demais *stakeholders*, buscar o entendimento sobre os objetivos do projeto e definir os requisitos que definem as funcionalidades do software.
- Planejamento: nesta atividade é definido um plano para a execução do projeto, ou seja, um guia que vai auxiliar a equipe durante o andamento do projeto, definindo as etapas e tarefas técnicas que precisam ser concluídas, riscos que podem afetar a conclusão do projeto, os recursos que serão utilizados durante o projeto, os produtos de trabalho e o cronograma do projeto.
- Análise e Modelagem: são definidos os modelos usados para uma melhor representação e entendimento dos requisitos do software. Os modelos são

usados pela equipe de desenvolvimento como base para implementação de software.

- **Construção:** implementação do software, onde o código fonte será escrito e testes serão executados visando garantir que o funcionamento do software esteja de acordo com o planejado e atenda aos objetivos dos *stakeholders*.
- **Entrega:** o software é entregue ao cliente, que faz sua avaliação e fornece *feedback* a respeito do produto.

2.2 Processo de Software e Padrões de Processos

Sommerville (2007) define o Processo de Software como um conjunto de atividades relacionadas que conduz a produção de um produto de software. Existem diversos processos de software, porém todos devem incluir quatro atividades fundamentais.

- 1. Especificação do Software:** define as funcionalidades do software;
- 2. Design e implementação do Software:** produção do software de acordo com a especificação;
- 3. Validação do Software:** o software deve ser validado para garantir o atendimento às necessidades do cliente;
- 4. Evolução do Software:** o software deve evoluir continuando atendendo as novas necessidades do cliente.

Processos de software são complexos e variam de acordo com diversos fatores, como o tipo do produto de software, tecnologia empregada, nicho de mercado, tamanho da equipe, entre outros. Não existe um processo perfeito, embora a busca pela melhoria do processo esteja na pauta de muitas empresas. O processo pode estar ultrapassado, ou não usa as boas práticas da engenharia de software.

A evolução das técnicas, métodos e ferramentas para construção de software, bem como a introdução de novas necessidades de software, introduz novos problemas e novos desafios em relação ao processo de software. Visando introduzir soluções comprovadas a estes problemas, muitas empresas adotam processos de software

baseados em Padrões de Processos. De acordo com Pressman (2009), os Padrões de Processos fornecem um modelo, um método consistente para descrever soluções para problemas dentro do contexto do processo de software.

A simples existência de um processo de software baseado em um padrão de processo não garante que o software será entregue com qualidade, no prazo e que ele atenderá às expectativas do cliente. É preciso avaliar o processo a fim de se averiguar se ele atende um conjunto de critérios básicos que tem se mostrado essenciais para se atingir as boas práticas da engenharia de software com sucesso.

Exemplos de padrões de processos são CMMI-DEV, MPS.BR e ISO/IEC 29110. Cada um destes modelos possui seu método de avaliação e certificação do processo de software. Estes três modelos serão apresentados no capítulo seguinte, e serão comparados sob a ótica das pequenas empresas.

2.3 Gerenciamento de Projetos

O gerenciamento de projetos de software é uma parte essencial da engenharia de software. Projetos precisam ser gerenciados porque o desenvolvimento de software está sujeito a restrições organizacionais e orçamentárias. O gerente de projetos deve trabalhar para garantir que o projeto supere estas restrições, atinja seus objetivos e que um produto de qualidade seja entregue no final. O bom gerenciamento não garante o sucesso do projeto, porém um mal gerenciamento geralmente resulta em falha do projeto (Sommerville, 2006).

Podemos afirmar que o gerenciamento de projetos é uma disciplina da engenharia de software, cujo objetivo é planejar, controlar, motivar equipe e controlar os recursos disponíveis de forma a atingir as metas do projeto.

De acordo com Sommerville, existem 5 atividades básicas que devem ser executadas por todos gerentes de projetos. Na atividade inicial, cujo nome é Planejamento do Projeto, o gerente de projetos é responsável por planejar, estimar esforço e definir as datas e fases do projeto, bem como designar a equipe responsável por cada tarefa.

A segunda atividade, denominada Reporte, é onde o andamento do projeto é reportado ao cliente, bem como a todas as partes interessadas dentro da companhia produtora do software. Este relatório de progresso deve ser descrito em linguagem apropriada, de acordo com o seu público alvo. Por exemplo, um relatório destino a área de negócios do cliente não deve entrar em detalhes técnicos de implementação de software.

O Gerenciamento de Riscos, terceira atividade proposta por Sommerville, define as tarefas de avaliação dos riscos que podem afetar o projeto, monitorar estes riscos e tomar ações quando problemas ocorrerem.

De acordo com o autor, o gerente de projetos é responsável por gerenciar o time de pessoas designadas para o projeto. Ele deve escolher a equipe e definir formas de afetar positivamente a performance desta equipe. Esta atividade é denominada Gerenciamento de Pessoas.

A quinta atividade, Definição da Proposta é o primeiro estágio em um projeto de software e envolve a criação da proposta e demais produtos de trabalho, que permitam a empresa de software ganhar o contrato de desenvolvimento junto ao seu cliente. A proposta contém os objetivos do projeto, estimativas de prazo e custo e informações sobre a equipe responsável pelo desenvolvimento do software.

Existem diversos modelos de gerenciamento de projetos, sendo que o modelo mais utilizado é o *Guide to the Project Management Body of Knowledge* (PMBok) publicado pelo *Project Management Institute* (PMI). Além disso, diversos padrões de processos propõem atividades de gerenciamento de projetos, como é o caso do CMMI-DEV, MR.MPS.SW e ISO/IEC 29110-4.

2.4 Requisitos de Software

De acordo com o Glossário da Engenharia de Software do IEEE (1990), os requisitos de software são definidos como:

- Uma condição ou uma capacidade requerida pelo usuário para resolver um problema ou atingir a um objetivo;
- Uma condição ou uma capacidade que precisa ser vista ou processada por um sistema ou componentes de sistema, de forma a satisfazer um contrato, um padrão, especificação ou outro documento imposto formalmente;
- Uma representação documentada de uma condição ou capacidade

A Engenharia de Requisitos é um subconjunto da Engenharia de Software que trata dos processos necessários para definir, documentar e manter os requisitos. Existem diversas técnicas para levantar e refinar os requisitos de software. Podem ser usados questionários, reuniões com os usuários e *stakeholders*, acompanhamento do processo entre outras técnicas. O papel responsável pela Engenharia de Requisitos dentro da empresa é o Analista (AN).

Os requisitos de software descrevem o que o sistema deve fazer, ou seja, as funcionalidades do software e suas limitações operacionais. Os requisitos refletem as necessidades do cliente por um sistema, que por sua vez deve servir a um determinado propósito. O processo de encontrar, analisar, documentar e validar estas funcionalidades e limitações é chamado de Engenharia de Requisitos (Sommerville, 2007).

Requisitos são tipicamente divididos em duas categorias, requisitos funcionais e requisitos não funcionais.

Os requisitos funcionais descrevem as funcionalidades ou comportamento do software, por exemplo como o sistema deve reagir a um determinado input, ou como um determinado cálculo deve ser executado.

Os requisitos não funcionais descrevem limitações ou critérios usados para julgar o funcionamento do software como um todo, ao invés de comportamentos e funcionalidades específicas. Como exemplo podemos citar o requisito não funcional de acessibilidade. Este requisito indica que o software deve possuir recursos que permitam ou facilitem o uso por portadores de deficiências.

2.5 Verificação e Validação

Testes software são recursos usados para se detectar erros no software. Erros podem ocorrer devido a uma falha durante a construção do software, por exemplo, divisão de uma variável por 0. Erros também podem ser reflexo de uma falha nas etapas de levantamento de requisitos e design, por exemplo um software deve calcular o salário líquido de um funcionário, porém a dedução dos impostos foi esquecida.

De acordo com Pressman (2009), os testes são elementos de Verificação e Validação, tópico este comumente representado como V&V. A Verificação garante que o software implementa uma função de forma correta. A Validação garante que o software produzido atende os requisitos do cliente.

De forma simplificada, Verificação responde à pergunta “estamos construindo o produto de forma correta? ”. Já a Validação responde à pergunta “estamos construindo o produto certo? ”.

2.6 Solicitações de Mudança

Solicitações de mudança podem ser caracterizadas como mudanças no produto ou mudanças no projeto. Mudanças no produto alteram o escopo do software, introduzindo alterações nos requisitos funcionais ou não funcionais do projeto. Estas alterações podem aumentar ou diminuir o escopo do produto. Já as mudanças no projeto introduzem mudanças nas atividades do projeto, como por exemplo a saída de um membro da equipe do projeto.

De acordo com o grupo IT Project Management (2007), existem 5 principais razões para as mudanças no escopo do projeto:

- 1. Um evento externo:** fatores fora do controle da equipe e do cliente (por exemplo, mudanças na legislação);
- 2. Um erro no escopo do produto:** possíveis omissões, imprecisões ou erros de cálculo relacionados ao produto ou ao projeto (por exemplo, regra de cálculo de imposto está errada);

3. Um erro no escopo do projeto: erros de estimativa ou planejamento do trabalho. Estas estimativas, geralmente feitas na fase inicial do projeto, podem apresentar erros como subestimar o tempo necessário para concluir cada tarefa ou falhas na divisão do trabalho em cada fase do projeto;

4. Uma mudança para agregar valor: são causadas por fatores que não podem ser previstos quando o escopo original está definido, mas se implementadas no escopo do projeto, podem melhorar o projeto ou torná-lo mais rentável (por exemplo, surgimento de uma nova tecnologia que facilita a implementação de um determinado requisito);

5. Um plano de contingência: um plano de contingência é aplicado de forma a mitigar riscos identificados em um projeto. Se o risco tem um impacto maior do que o previsto, uma mudança no escopo do projeto pode ser necessária.

2.7 Considerações do Capítulo

Neste capítulo, foram apresentados conceitos relacionados a Engenharia de Software e boas práticas para desenvolvimento de Software. No capítulo a seguir, dois padrões de desenvolvimento e uma norma internacional serão apresentados descritos. Somente um será escolhido para o estudo de caso, apresentado no capítulo 4.

3. BOAS PRÁTICAS PARA DESENVOLVIMENTO DE SOFTWARE

Para a engenharia de software, um padrão (ou processo) de desenvolvimento é uma coleção de atividades e tarefas que devem ser executadas de forma sistemática para que, ao final, se tenha um software que atenda aos requisitos de qualidade e funcionalidades definidas pelos *stakeholders*. Os padrões de desenvolvimento funcionam como frameworks onde a empresa pode criar ou adaptar seu próprio processo com base em padrões pré-estabelecidos. Além disso, alguns destes padrões são certificáveis, atestando a qualidade do processo da empresa.

Para este trabalho, foram selecionados o CMMI-DEV, MPS.BR-SW e uma nova norma internacional recentemente introduzida, sob o nome ISO/IEC 29110-4. Nos próximos tópicos estes padrões serão detalhados e será estudada a viabilidade de sua adoção em pequenas empresas.

3.1 CMMI-DEV

O CMMI-DEV (*Capability Maturity Model Integration*) é um modelo de maturidade para melhoria de processo de desenvolvimento de produtos e serviços de software. O processo foi criado pela SEI (*Software Engineering Institute*) da universidade *Carnegie Mellon* e publicado inicialmente no ano 2000. Atualmente o órgão responsável pelo modelo é o *CMMI Institute*, independente, porém ainda ligado à Universidade *Carnegie Mellon*. Desde sua publicação inicial, esta organização fez diversas atualizações ao padrão, lançando em 2010 a versão mais recente, conhecida como CMMI-DEV v1.3.

Atualmente o CMMI-DEV possui um grande número de empresas adeptas no mercado, tendo adquirido um status de “padrão *de facto*” para a indústria de software. Um relatório publicado em 2014 pelo *CMMI Institute* afirma que milhares de empresas ao redor do globo, de diferentes setores industriais em 94 países adotaram as práticas do CMMI-DEV para melhorar seu desempenho, e foram avaliadas quanto a sua capacidade e maturidade usando modelos CMMI. De acordo com este relatório, 1.841 usuários de CMMI foram convidados a avaliar qual a possibilidade de se recomendar o uso do modelo a amigos e colegas. De acordo com os resultados do questionário,

45% dos entrevistados afirmaram que indicariam o modelo, citando como motivo “ganhos comprovados em desempenho”, enquanto 54% dos entrevistados afirmam que indicariam o modelo pois o veem como “o padrão *de facto* para melhoria de processos”. (CMMI Institute)

De acordo com o guia CMMI-DEV v1.3, o modelo foi desenvolvido de acordo com as melhores práticas relacionadas aos processos necessários ao desenvolvimento de software, cobrindo todo seu ciclo de vida, desde etapas iniciais de concepção e modelagem, até as etapas posteriores de entrega e manutenção. A adoção do modelo CMMI-DEV por uma empresa deve se iniciar pela escolha da melhor abordagem de implementação, pois o CMMI-DEV permite a adoção contínua ou por etapas.

Na adoção contínua, é possível que a empresa utilize uma ordem de melhoria que seja mais aderente aos objetivos da empresa a curto prazo. Desta forma, a empresa define uma área de processo a ser melhorada individualmente e também define a qual nível de capacidade se pretende chegar. Caso escolha a adoção por etapas, a empresa deve atingir a um nível de maturidade em todo processo, obedecendo uma sequência pré-determinada para melhoria baseada em estágios. Cada estágio atingido serve de base para o próximo, conforme ilustrado na tabela 1.

Tabela 1 - Níveis de maturidade do CMMI-DEV

Nível	Nome	Descrição
Nível 1	Inicial	Os processos não são bem definidos, sucesso depende da capacidade individual da equipe
Nível 2	Gerenciado	Os processos são executados de acordo com a política da organização, produzem saídas controladas, é monitorado, controlado e revisado.
Nível 3	Definido	Os processos são bem caracterizados e entendidos, e são descritos em normas, procedimentos, ferramentas e métodos. O conjunto de processos padrão da organização é definido para futura revisão e melhoria.

Nível 4	Quantitativamente gerenciado	A organização define objetivos quantitativos de qualidade e desempenho de processos, usando-os como critério para gestão de projetos.
Nível 5	Em otimização	A organização melhora continuamente seus processos com base no entendimento quantitativo de seus objetivos de negócio e necessidades de desempenho.

Fonte: CMMI for Development, Version 1.3

O modelo CMMI-DEV é estruturado a partir de Áreas de Processo. As áreas de processos descrevem os aspectos do desenvolvimento de software que serão cobertos pelos processos organizacionais. Em outras palavras, são um conjunto de práticas que satisfazem as metas julgadas necessárias para que sejam feitas significativas melhorias ao processo. As áreas de processos são listadas na tabela 2.

Tabela 2 - Áreas de processo do CMMI-DEV

Nível de maturidade	Área de processo
Nível 2	CM - Gestão da configuração MA - Mensuração e análise PMC - Monitoramento e controle do projeto PP - Planejamento do projeto PPQA - Garantia da qualidade do processo e do produto REQM - Gestão dos requisitos SAM - Gestão de fornecedores
Nível 3	DAR - Análise de decisão e resolução IPM - Gerenciamento de projeto integrado OPD - Definição do processo organizacional OPF - Foco do processo organizacional OT - Treinamento organizacional PI - Integração do produto

	RD - Desenvolvimento dos requisitos RSKM - Gestão do risco TS - Solução técnica VAL - Validação VER - Verificação
Nível 4	OPP - Performance do processo organizacional QPM - Gestão quantitativa de projetos
Nível 5	CAR - Análise de causa e resolução OPM - Gestão da performance organizacional

Fonte: CMMI for Development, Version 1.3 (2010)

3.2 MPS.BR-SW

Fruto de uma iniciativa brasileira e criado em dezembro de 2003, o MPS-BR (Melhoria de Processo do Software Brasileiro) foi idealizado com o objetivo de proporcionar a melhoria do processo de software e serviços a médio e longo prazos no país. De acordo com o Guia Geral MPS de Software (2012), o modelo é adequado ao perfil de diversas empresas, com ênfase em micro, pequenas e médias empresas. O modelo é coordenado pela SOFTEX (Associação para Promoção da Excelência do Software Brasileiro) e conta com o apoio do Ministério da Ciência, Tecnologia e Inovação.

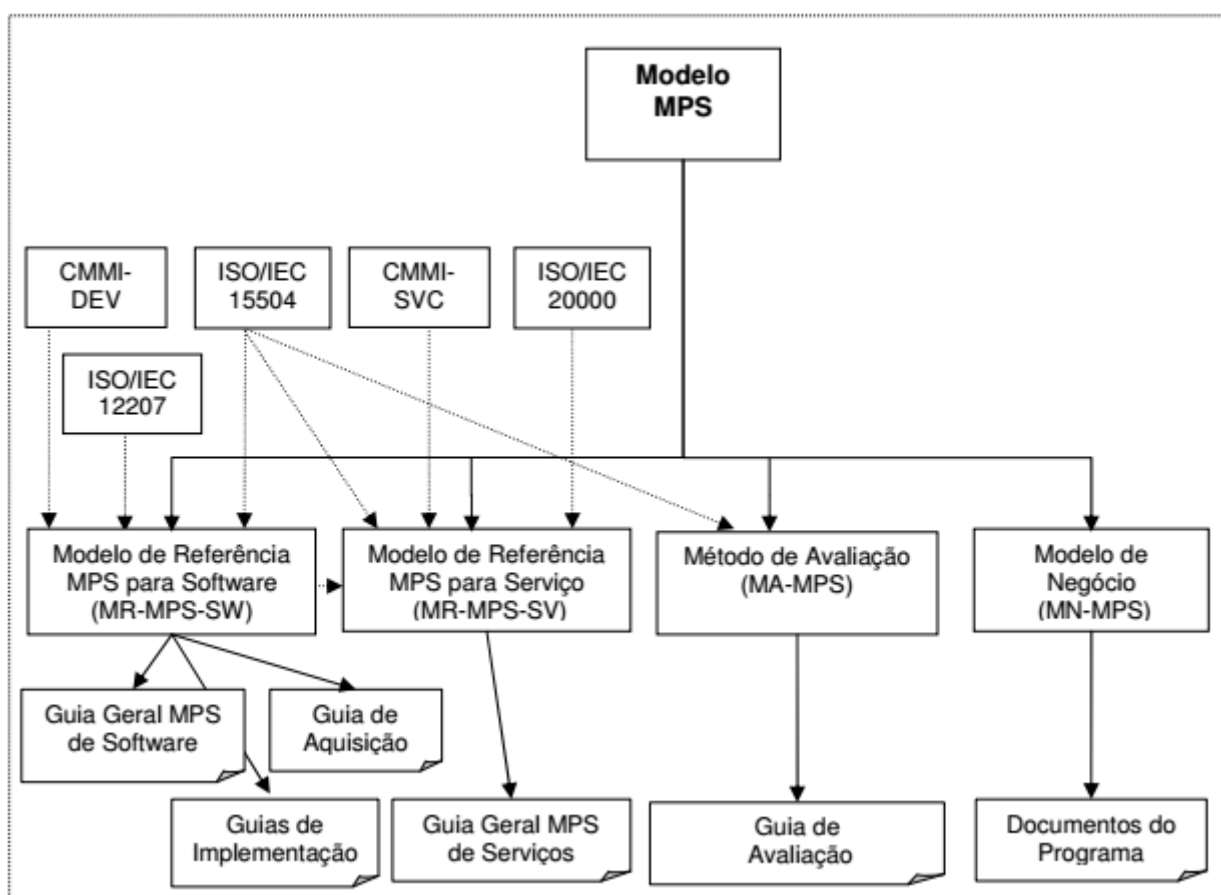
Para o modelo de referência MPS-BR, qualidade é fator crítico de sucesso para a indústria de software. A busca pela competitividade através da qualidade é fator fundamental pois implica na busca por melhoria na qualidade dos produtos de software e serviços correlatos bem como dos processos de elaboração conceitual, produção e distribuição de software. Portanto, o modelo foi concebido de forma a buscar compatibilidade com os padrões de qualidade aceitos internacionalmente, e que tenha como pressuposto o aproveitamento de padrões e modelos de melhoria de processo já disponíveis atualmente.

O programa MPS é composto por 4 componentes, sendo eles o Modelo de Referência para Software (MR-MPS-SW), o Modelo de Referência para Serviços (MR-MPS-SV),

o Método de Avaliação (MA-MPS) e o Modelo de Negócio (MN-MPS). Para este trabalho, será abordado somente o MR-MPS-SW. Foram usados como base para a elaboração do modelo as seguintes normas ISO/IEC:

- *ISO/IEC 15504 Information technology — Process assessment*
- *ISO/IEC 12207 Systems and software engineering — Software life cycle processes.*

Figura 2 - Componentes do modelo MPS



Fonte: Guia Geral MPS de Software (2012)

O modelo de referência MPS.BR-SW define níveis de maturidade, uma combinação entre processos e sua capacidade. Os níveis de maturidade estabelecem graus de evolução de processos, ou seja, degraus que precisam ser percorridos de forma sequencial pela organização que deseja implementar a melhoria de processos. O MPS.BR-SW define 7 níveis de maturidade:

Tabela 3 - Níveis de maturidade MPS

Nível de maturidade	Nomenclatura
A	Em otimização
B	Gerenciado quantitativamente
C	Definido
D	Largamente definido
E	Parcialmente definido
F	Gerenciado
G	Parcialmente gerenciado

Fonte: Guia Geral MPS de Software (2012)

De acordo com a tabela 3, a escala de maturidade se inicia no nível G e progride até o nível A. Cada um dos níveis possui um perfil de processos que definem onde a organização deve focar esforços para obter a melhoria de processos. O progresso de um nível a outro mais elevado se obtém quando a organização consegue atender os propósitos e todos os resultados esperados dos respectivos processos e os resultados dos atributos de processo estabelecidos para aquele nível.

Atributo de processo é uma característica mensurável da capacidade do processo aplicável a qualquer processo. A tabela 4 exibe a relação de atributos de processo.

Tabela 4 - Atributos de processo

AP 1.1	O processo é executado
AP 2.1	O processo é gerenciado
AP 2.2	Os produtos de trabalho do processo são gerenciados
AP 3.1	O processo é definido
AP 3.2	O processo está implementado
AP 4.1	O processo é medido
AP 4.2	O processo é controlado
AP 5.1	O processo é objeto de melhorias incrementais e inovações
AP 5.2	O processo é otimizado continuamente

Fonte: Guia Geral MPS de Software (2012)

A tabela a seguir apresenta os processos presentes em cada nível de maturidade do padrão MPS.BR-SW, bem como seus respectivos atributos de processo.

Tabela 5 - Níveis de maturidade MPS.BR-SW

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2 , AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI Desenvolvimento para Reutilização – DRU Gerência de Decisões – GDE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
D	Verificação – VER Validação – VAL Projeto e Construção do Produto – PCP Integração do Produto – ITP Desenvolvimento de Requisitos – DRE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	Gerência de Projetos – GPR (evolução) Gerência de Reutilização – GRU Gerência de Recursos Humanos – GRH Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional – AMP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
F	Medição – MED Garantia da Qualidade – GQA Gerência de Portfólio de Projetos – GPP Gerência de Configuração – GCO	AP 1.1, AP 2.1 e AP 2.2

	Aquisição – AQU	
G	Gerência de Requisitos – GRE Gerência de Projetos – GPR	AP 1.1 e AP 2.1

Fonte: Autor

3.3 ABNT NBR ISO/IEC 29110-4

A busca pela qualidade no desenvolvimento de software é um assunto cada vez mais importante para as companhias de diversos tamanhos. Tal interesse despertou a atenção da ISO (*International Organization for Standardization*) e do IEC (*Internacional Eletrotecnical Commission*) em criar normas e padrões que sirvam de referência para que empresas possam criar seus processos de software de forma organizada e padronizada.

Em 1987 ambas entidades criaram um JTC 1(Comitê Técnico Conjunto) visando o desenvolvimento de normas e padrões específicos para a área da tecnologia da informação. O JTC 1 por sua vez é dividido em diversos subcomitês, cada um responsável por uma determinada área de conhecimento. O subcomitê 7 (SC 7) é responsável pelos padrões internacionais referentes a engenharia de software e sistemas.

De acordo com pesquisas e avaliações realizadas pelo SC 7, ficou claro que a maioria dos padrões internacionais não atendem às necessidades das micro organizações. Segundo o SC 7, uma VSE caracteriza uma organização (empresa, departamento, projeto) que possui até 25 pessoas. A implantação das normas existentes é muito complexa e poucas VSEs possuem estrutura em termos de recursos humanos e financeiros para arcar com a adoção de um padrão internacional.

As VSEs em geral não são reconhecidas internacionalmente como produtoras de software e sistemas de qualidade, limitando a área de atuação de uma VSE e excluindo-a de algumas atividades econômicas. VSEs podem desenvolver ou prestar manutenção em sistemas ou componentes de software que são usados em sistemas

maiores. Portanto o reconhecimento como fornecedores de sistemas de alta qualidade muitas vezes é mandatório.

Com objetivo de melhorar o reconhecimento das VSEs, foi criado um conjunto de guias e normas de acordo com um conjunto de características deste tipo de organização. Os guias baseiam-se em conjuntos apropriados de elementos de normas, denominados Perfis de VSEs. O propósito de um perfil de VSE é definir um subconjunto de Normas Internacionais relevante para o contexto da VSE, como, por exemplo, processos e resultados da ISO/IEC 12207 e produtos da ISO/IEC 15289. Dessa forma se aplicam as melhores práticas de engenharia de software um formato leve e pensado para se adequar a realidade das VSEs.

A ISO/IEC 29110: *Systems and Software Lifecycle Profiles and Guidelines for Very Small Entities* (VSEs) é uma norma voltada a pequenas empresas e/ou departamentos, com no máximo 25 pessoas. A norma foi criada para atender à necessidade por padrões leves de sistemas de engenharia de software.

No Brasil, a ABNT (Associação Brasileira de Normas Técnicas) adaptou a norma, portanto este trabalho é baseado na norma ABNT NBR ISO/IEC 29110-4-1:2012 Engenharia de Software — Perfis de ciclo de vida para micro organizações (VSEs). Para efeitos de simplicidade, a partir deste ponto ela será denominada ISO/IEC 29110-4.

A norma define processos, atividades e objetivos necessários para desenvolvimento de software, visando pequenas empresas que não possuem a quantidade ideal de recursos humanos e financeiros para adoção de padrões mais robustos e criados visando grandes empresas, como o CMMI-DEV.

A ISO/IEC 29110 se aplica a VSEs, ou seja, empresas/organizações/departamentos ou projetos de até 25 pessoas que desejam ser reconhecidos como empresas que desenvolvem software com qualidade. A norma está dividida em 5 partes, cada uma dedicada a um público alvo, como mostra a tabela 6. A parte 4 (29110-4-n) será usada como base deste trabalho, pois ela é a parte auditável e contém os requisitos normativos para certificação.

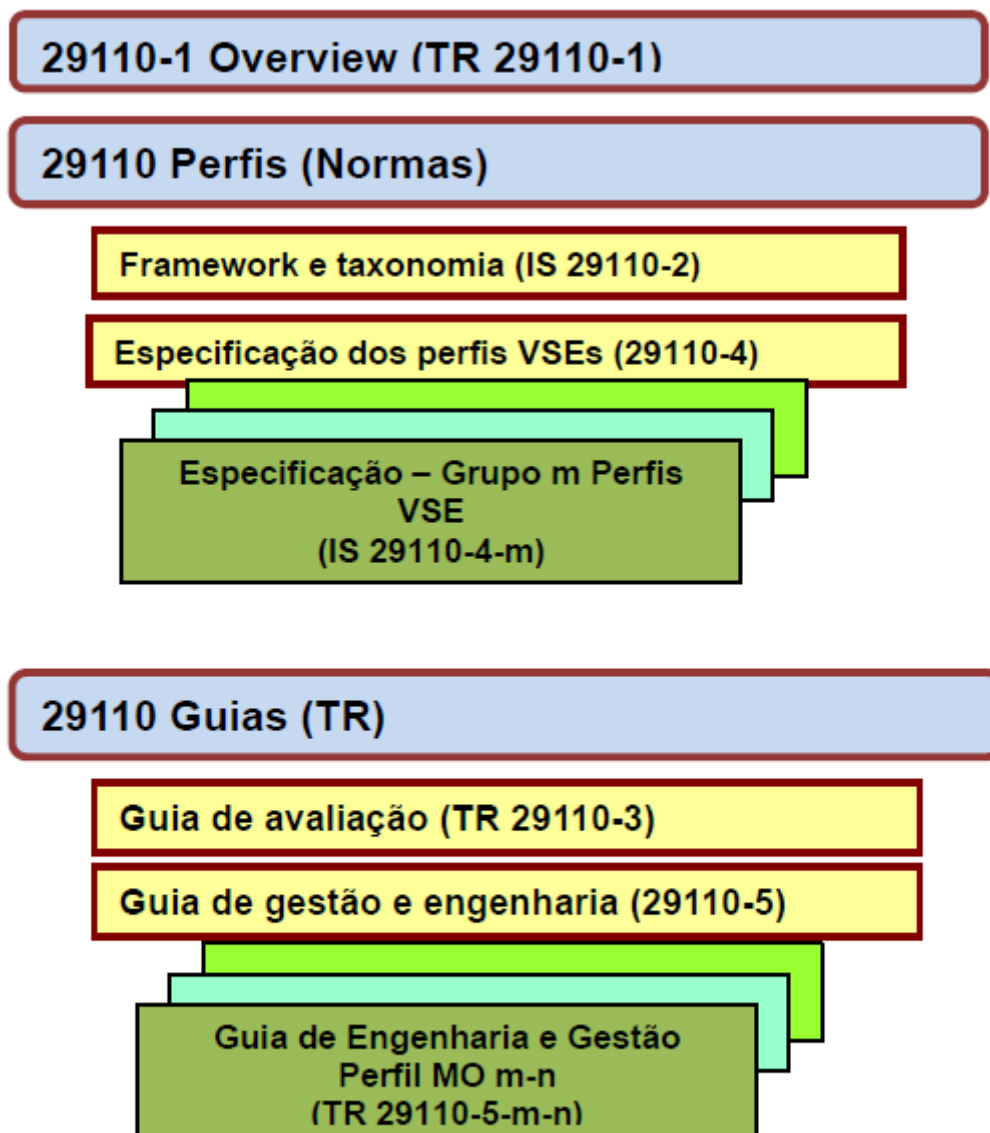
Tabela 6 - ISO/IEC 29110 por público alvo

ISO/IEC 29110	Título	Público
Parte 1	Visão geral	VSEs, avaliadores, produtores de normas, fornecedores de ferramentas e fornecedores de metodologias
Parte 2	Estrutura e taxonomia	Produtores de normas, fornecedores de ferramentas e fornecedores de metodologia. Não é dirigida a VSEs.
Parte 3	Guia de avaliação	Avaliadores e VSEs
Parte 4	Especificação de perfis	Produtores de normas, fornecedores de ferramentas e fornecedores de metodologia. Não é dirigida a VSEs.
Parte 5	Guia de engenharia e gestão	VSEs

Fonte: ABNT NBR ISO/IEC 29110-4 (2012)

As 5 partes da norma se agrupam como Overview (visão geral, introdução), Perfis (descrição dos perfis básicos e requisitos normativos) e Guias (também chamados de *Technical Report* – relatório técnico, um guia de implementação sem valor normativo). A figura 3 apresenta este agrupamento.

Figura 3 - Estrutura da norma ISO/IEC 29110



TR=Relatório Técnico

IS=Norma Internacional

A necessidade de melhoria de processos é uma questão estratégica para os negócios da empresa, motivada pela busca por lucros e melhoria da competitividade. Uma vez identificada a necessidade estratégica pela busca de melhoria no processo de desenvolvimento de software, uma organização deve adotar padrões de engenharia de software. Padrões de engenharia de software focam em processos e aspectos do produto. Eles contêm requisitos formais que servem como abordagens constantes de desenvolvimento de software. Padrões de Engenharia de Software possuem diversos objetivos (ABNT NBR ISO/IEC 29110-4, 2012):

- Fornece um framework comum e vocabulário para os praticantes do desenvolvimento de software.
- Fornece um framework para acordo entre duas partes (cliente x desenvolvedor de software).
- Avaliar e melhorar a competência no desenvolvimento de software.
- Facilitar avaliação do processo de desenvolvimento e do produto.

Padrões possuem partes normativas, e em alguns casos, partes informativas. As partes normativas de um padrão de engenharia de software são usadas como requisitos para avaliação de conformidade. As partes informativas contêm informações que complementam ou facilitam a compreensão ou uso das partes normativas.

Devido ao tamanho reduzido das VSEs, a adoção de padrões de engenharia de software pode se tornar um processo custoso e inviável. Desta forma, o JTC1, responsável pela criação da ISO/IEC 29110 optou pela criação de subconjuntos dos principais padrões internacionais de engenharia de software, denominando-os “perfis de ciclo de vida”. Os padrões usados na definição de Perfis são:

- ISO/IEC 12207:2008 Systems and software engineering -- Software life cycle processes;
- ISO/IEC 15288:2008 Systems and software engineering -- System life cycle processes;

- ISO/IEC 15289:2011 Systems and software engineering -- Content of life-cycle information products;

Como dito anteriormente, a principal característica das empresas alvo da ISO/IEC 29110 é o tamanho. Porém, existem outros aspectos e características que podem afetar a seleção de um determinado perfil a uma organização, como o modelo de negócios, o desenvolvimento de sistemas de alto risco e alta disponibilidade, incertezas ambientais e demais variáveis. Portanto, criar um perfil que se adequa a cada uma das possíveis combinações entre subconjuntos de padrões de engenharia de software e as variações entre cada organização resultaria em um infinito número de combinações, tornando a este trabalho inviável (ABNT NBR ISO/IEC 29110-4, 2012).

De forma a resolver esta questão, a equipe responsável pela criação da norma ISO/IEC 29110 propôs a criação de “Grupos de Perfis Genéricos”. Desta forma, perfis semelhantes serão agrupados para que seja possível sua aplicação a mais de uma categoria de empresa. Cada grupo de perfil genérico receberá seu próprio guia de implementação, detalhando todos os processos, atividades e controles que devem ser implementados para que a organização seja certificada. A tabela 7 exibe todos os perfis genéricos publicados até o momento.

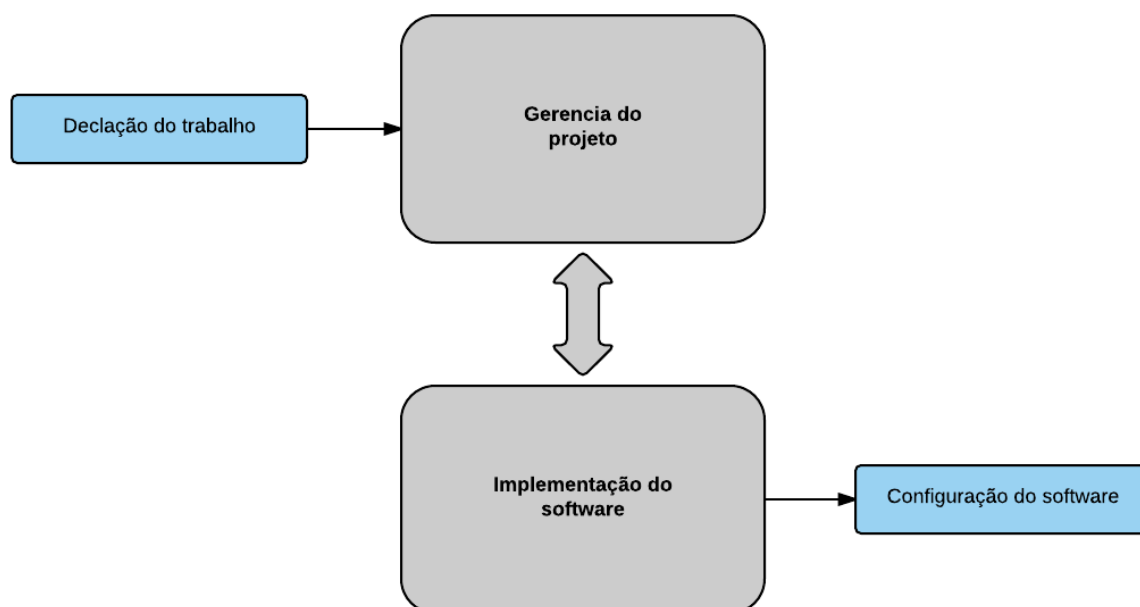
Tabela 7 - Relação de Grupos de Perfis Genéricos

Código do documento	Nome	Descrição
<u>ISO/IEC TR 29110-5-1-1</u>	<i>Entry profile</i>	Perfil genérico específico para empresas <i>startups</i>
<u>ISO/IEC TR 29110-5-1-2</u>	<i>Basic profile (2011)</i>	Primeira publicação do guia de implementação da norma. Voltado para empresas com produtos de software não críticos.
<u>ISO/IEC TR 29110-5-6-2</u>	<i>Basic profile (2014)</i>	Atualização do guia de implementação volta a empresas desenvolvedoras de sistemas não críticos
-	<i>Intermediate profile</i>	Publicação esperada para 2016
-	<i>Advanced profile</i>	Publicação esperada para 2016

Fonte: Autor

A ISO/IEC 29110-4 propõe um perfil básico que atende a maioria das empresas. O perfil básico é composto pelas etapas de declaração do trabalho, gerência de projetos, implementação de software e configuração do software. Desta forma a norma fornece os processos de Gerencia de Projetos (*Project Management - PM*) e Implementação de Software (*Software Implementation - SI*) integram práticas baseadas na seleção de elementos das normas ABNT NBR ISO/IEC 12207:2008, *Systems and Software engineering - Software life cycle processes* e ISO/IEC 15289:2006, *System and software engineering - Content of systems and software life cycle process information products*.

Figura 4 - O processo da norma ISO/IEC 29110



Fonte: ABNT NBR ISO/IEC 29110-4 (2012)

De acordo com a visão geral da norma ISO/IEC 29110-4, os principais benefícios obtidos através de implementação da norma ISO/IEC 29110 estão nos seguintes aspectos:

- Um conjunto acordado de requisitos do projeto e produtos esperados é entregue ao Cliente;
- Um processo de gerenciamento disciplinado que fornece visibilidade do projeto e ações corretivas para os problemas e desvios do projeto é executado;

- Um processo sistemático de implementação de software que satisfaz as necessidades do Cliente e que garante produtos de qualidade é seguido;

Cada processo possui como propósito o estabelecimento e realização de forma sistemática de suas tarefas, cumprindo desta forma os objetivos do projeto dentro da qualidade, tempo e custos esperados. Portanto, cada processo (PM e SI) possui um conjunto de Objetivos (por exemplo: PM.O1 a PM.O7) e atividades (PM.1.1 a PM.4.2). Atividades representam um conjunto de tarefas coesas, como um requisito, uma recomendação ou ação cuja intenção seja contribuir para realização de um ou mais objetivos do processo. Objetivos são metas específicas para garantir o acompanhamento do propósito do processo. Cada objetivo é seguido por um quadro que inclui uma lista dos processos escolhidos da ABNT NBR ISO/IEC 12207:2009 para o perfil básico e seus resultados relacionados ao objetivo.

A empresa que deseja adotar a norma ISO/IEC 29110-4 deve possuir os seguintes requisitos:

- 2 processos, um para gestão de projetos e um para implementação de software
- 4 atividades de gestão de projetos
- 7 objetivos de gestão de projetos
- 6 atividades de implementação de software
- 7 objetivos de implementação de software
- 22 produtos de trabalho

Todos os projetos da organização devem ser realizados com os itens acima. Ela deve executar as **atividades**, atender os **objetivos** e produzir os **produtos de trabalho** obrigatórios.

De acordo com a especificação da norma ISO/IEC 29110-4, o processo Gerência de Projeto (PM) usa a Declaração de Trabalho do Cliente para elaborar o Plano do Projeto. As tarefas de avaliação e controle do projeto do PM comparam o progresso do projeto frente ao Plano do Projeto e ações são tomadas para eliminar desvios ou incorporar mudanças ao Plano do Projeto. A atividade de encerramento do projeto do PM entrega a Configuração do Software, produzida pelo Processo Implementação de

Software (SI), e obtém o aceite do Cliente para formalizar o final do projeto. Um Repositório do Projeto é estabelecido para armazenar os produtos de trabalho e para controlar suas versões durante o projeto.

A execução do Processo SI é orientada pelo Plano do Projeto. O processo SI começa com uma atividade de iniciação da revisão do Plano do Projeto. O Plano do Projeto guia a execução das atividades de análise dos requisitos de software, Projeto de arquitetura e detalhamento do software, construção do software, integração e teste e entrega do produto.

A tabela 8 apresenta a relação de papéis participantes do processo especificado pela norma.

Tabela 8 - Relação de papéis

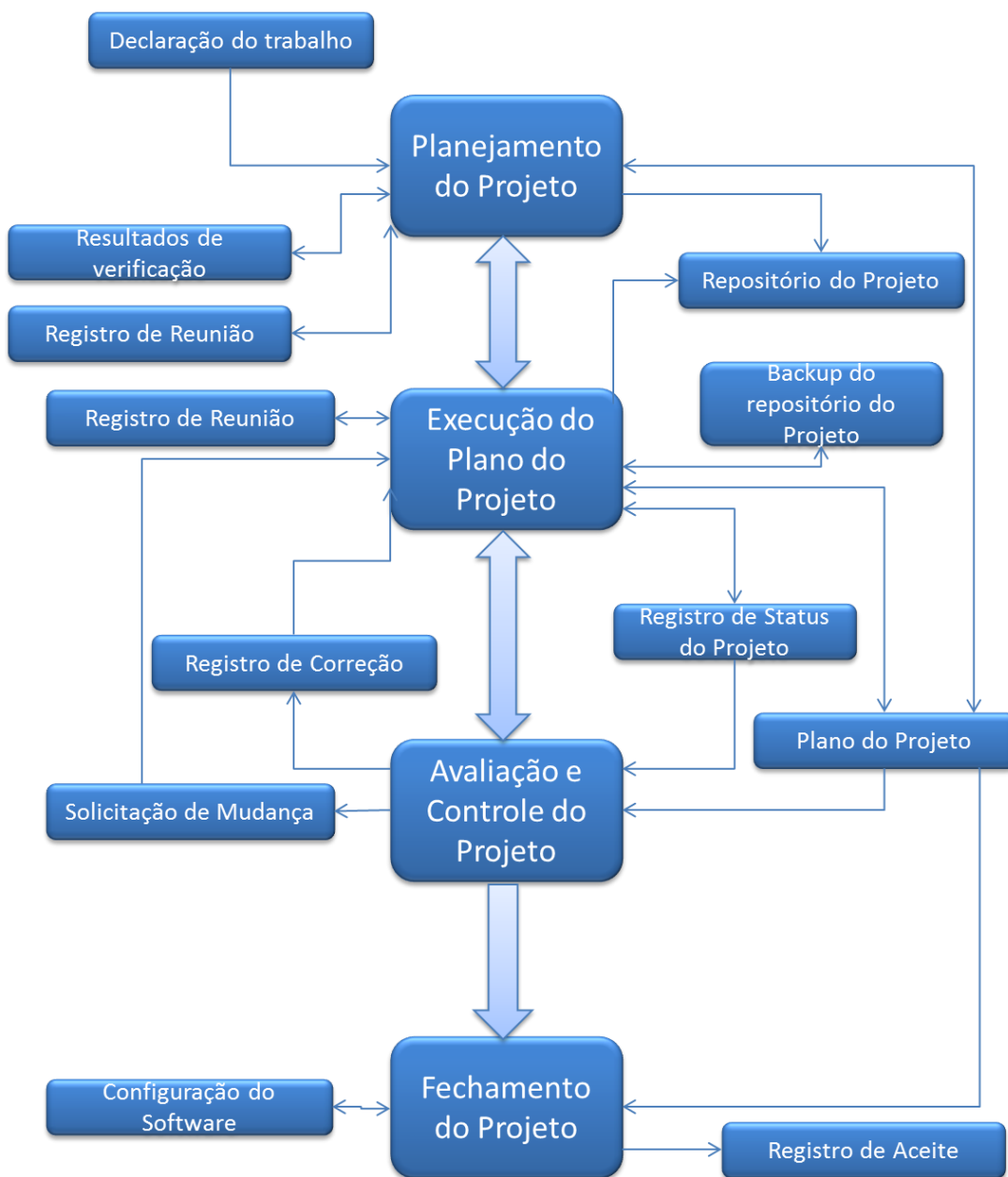
Papel	Descrição
AN	Analista (<i>Analyst</i>)
CUS	Cliente (<i>Customer</i>)
DES	Projetista (<i>Designer</i>)
PM	Gerente de Projetos (<i>Project Manager</i>)
PR	Programador (<i>Programmer</i>)
TL	Líder Técnico (<i>Technical Leader</i>)
WT	Equipe de trabalho (<i>Work Team</i>)

Fonte: ABNT NBR ISO/IEC 29110-4 (2012)

3.3.1 Processo de Gerenciamento de Projetos (PM)

O objetivo do processo de gerencia de projetos é definir e acompanhar as atividades necessárias à implementação de software de forma a entregar um produto que atenda às necessidades do cliente, qualidade, prazo e custo. A figura 5 exibe todas as atividades e tarefas necessárias ao processo PM.

Figura 5 - Atividades do processo de gerenciamento de projetos (PM)



O processo de gerência do projeto (PM - *Project Management*) possui as seguintes atividades:

- PM.1: Planejamento do projeto
- PM.2: Execução do plano do projeto
- PM.3: Avaliação e controle do projeto
- PM.4: Encerramento do projeto

O processo de gerencia do projeto (PM) possui os seguintes objetivos:

- PM.O1: O plano de projeto é desenvolvido de acordo com a declaração de trabalho e validado com o cliente;
- PM.O2: O progresso do projeto é monitorado com base no plano de projeto e registrado no registro de status do progresso;
- PM.O3: As solicitações de mudança são tratadas através da sua recepção e análise;
- PM.O4: São conduzidas reuniões de avaliação com a equipe de trabalho e o cliente;
- PM.O5: Riscos são identificados quando aparecem e durante o desenvolvimento do projeto;
- PM.O6: Uma estratégia de controle de versão de software é desenvolvida;
- PM.O7: A garantia da qualidade de software é realizada para assegurar que os produtos e processos de trabalho cumpram o plano do projeto e a especificação dos requisitos;

A tabela 9 relaciona atividades e objetivos do processo PM.

Tabela 9 - Atividades e objetivos do processo PM

PM – Atividade	PM – Objetivos	Descrição
PM.1 - Planejamento do projeto	PM.O1 - Planejamento do Projeto PM.O5 - Gerenciamento de Riscos PM.O6 - Gestão de Configuração de Software PM.O7 - Garantia de Qualidade de Software	- Declaração de trabalho revisada; - Ciclo de vida - Estimativa de esforço, tempo e custo;
PM.2 – Execução do Plano de Projeto	PM.O2 - Avaliação e Controle do Projeto PM.O3 - Análise de Requisitos de Software PM.O4 - Revisão de Software PM.O5 - Gerenciamento de Riscos PM.O7 - Garantia de Qualidade de Software	- Registrar status do projeto; - Avaliação de impacto de mudança de escopo;
PM.3 – Avaliação e Controle de Projeto	PM.O2 - Avaliação e Controle do Projeto	- Avalia a realização do plano do projeto frente aos objetivos; - Análise e revisão dos riscos do projeto;
PM.4 – Encerramento de Projeto	PM.O2 - Avaliação e Controle do Projeto	- Liberação dos produtos conforme Instruções de entrega; - Suporte; - Registro de aceite;

Fonte: ABNT NBR ISO/IEC 29110-4 (2012)

As atividades do processo PM possuem produtos de saída, que são produtos gerados pelo processo e seus respectivos destinos, que podem ser outro processo ou uma entidade externa do projeto, como o Cliente ou Gerência Organizacional.

A tabela 10 documenta dos produtos de trabalho do processo PM, qualificando-os como produtos de saída ou produtos internos.

Tabela 10 - Produtos de trabalho do processo PM

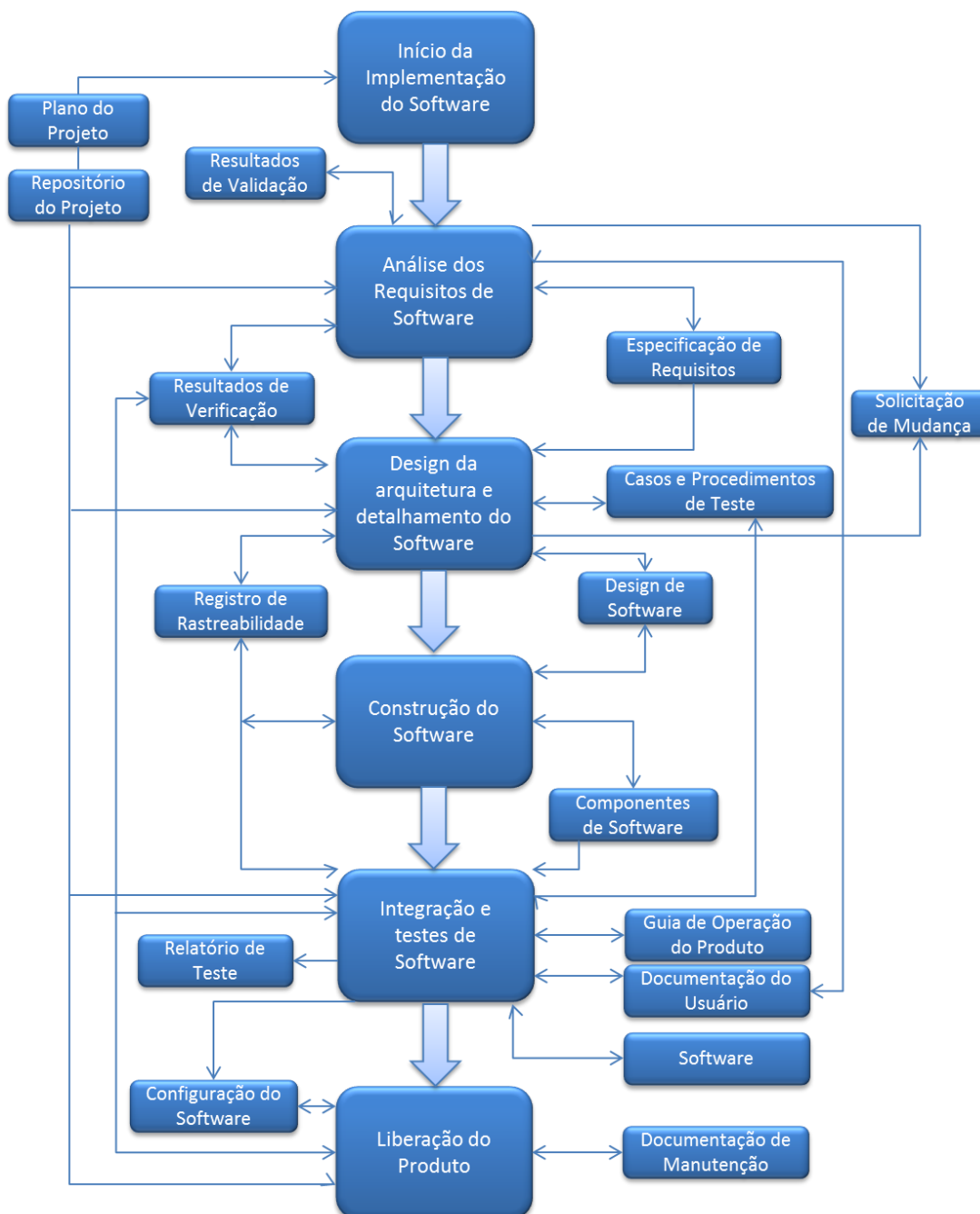
	Produto de trabalho	Produto de saída	Produto Interno
1	Plano do projeto	x	
2	Registro de aceite	x	
3	Solicitação de mudança		x
4	Repositório do projeto	x	
5	Backup do repositório do projeto	x	
6	Registro de reunião	x	
7	Registro de correção		x
8	Registro de status do projeto		x

Fonte: ABNT NBR ISO/IEC 29110-4 (2012)

3.3.2 Processo de Implementação de Software (SI)

O objetivo do processo de Implementação de Software é executar sistematicamente as atividades de desenvolvimento de software, ou seja, análise, design, construção, integração e testes, de acordo com os requisitos especificados.

Figura 6 - Atividades do processo de implementação de software (SI)



Fonte: ABNT NBR ISO/IEC 29110-4 (2012)

O processo de implementação de software (SI - *Software Implementation*) possui as seguintes atividades:

- SI.1: Início da implementação de software
- SI.2: Análise de requisitos de software
- SI.3: Projeto de arquitetura e detalhamento de software
- SI.4: Construção do software
- SI.5: Integração e testes do software
- SI.6: Entrega do produto

O processo de implementação de software (SI) possui os seguintes objetivos:

- SI.O1: Tarefas das atividades são realizadas através da execução do Plano do Projeto corrente.
- SI.O2: Requisitos de Software são definidos, analisados quanto à validade e testabilidade, aprovados pelo Cliente, incluídos em *baselines* e comunicados.
- SI.O3: O projeto de arquitetura e seu detalhamento são elaborados e incluídos em *baseline*. Ele descreve os Componentes de Software e suas interfaces internas e externas. A consistência e rastreabilidade aos requisitos do software são estabelecidas. SI.O4: Componentes de Software definidos no projeto são desenvolvidos. Testes unitários são definidos e realizados para verificar consistência com requisitos e o projeto. A rastreabilidade é estabelecida entre os requisitos e o projeto.
- SI.O5: O software é produzido realizando-se a integração dos Componentes de Software e verificado através de Casos de Teste e Procedimentos de Teste. Os resultados são armazenados no Relatório de Teste. Defeitos são corrigidos e consistência e rastreabilidade do projeto do software são estabelecidas.
- SI.O6: Uma Configuração de Software, que atende à Especificação dos Requisitos como acordado com o Cliente, a qual inclui documentação de usuário, manutenção e operação, é integrada, incluída em *baseline* e armazenada no Repositório do Projeto. Necessidades de mudança na Configuração do Software são detectadas e solicitações de mudança relacionadas são iniciadas.

- SI.O7: Tarefas de Verificação e Validação de todos os produtos de trabalho requeridos são realizadas usando os critérios definidos para garantir a consistência entre produtos de entrada e saída em cada atividade. Defeitos são identificados e corrigidos; registros são armazenados nos Resultados de Verificação/Validação.

A tabela 11 relaciona atividades e objetivos do processo SI.

Tabela 11 - Atividades e objetivos do processo SI

SI - Atividade	SI - Objetivos	Descrição
SI.1 - Início da Implementação do Software	SI.O1 - Execução do plano do projeto	- Revisão do plano do projeto; - Estabelecimento do ambiente de desenvolvimento;
SI.2 - Análise de Requisitos do Software	SI.O2 - Análise de Requisitos de Software SI.O6 - Fornecimento SI.O7 - Verificação e validação de Software	- Levantamento e validação dos requisitos; - Aprovação do cliente;
SI.3 - Projeto de Arquitetura e Detalhamento do Software	SI.O3 - Projeto de Arquitetura de Software SI.O6 - Fornecimento SI.O7 - Verificação e validação de Software	- Projeto arquitetural; - Detalhamento dos componentes do software; - Casos de teste;
SI.4 - Construção do Software	SI.O4 - Construção de Software SI.O6 - Fornecimento SI.O7 - Verificação e validação de Software	- Codificação; - Testes; - Rastreabilidade entre componentes e projeto de software;

SI.5 - Integração e Testes do Software	SI.O5 - Integração de Software SI.O6 - Fornecimento SI.O7 - Verificação e validação de Software	- Integrar componentes de software; - Entendimento e realização dos casos de teste; - Rastreabilidade dos requisitos e do projeto ao produto de software integrado; - Documentação e manutenção verificada;
SI.6 - Entrega do Produto		

Fonte: Autor

A tabela 12 documenta dos produtos de trabalho do processo SI, qualificando-os como produtos de saída ou produtos internos.

Tabela 12 - Produtos de trabalho do processo SI

	Produto de trabalho	Produto de saída	Produto Interno
1	Especificação de requisitos	x	
2	Software	x	
3	Componentes de software	x	
4	Configuração de software	x	
5	Projeto (design) de software	x	
6	Documentação do usuário do software	x	
7	Declaração de trabalho	x	

8	Casos de teste e procedimento de testes	x	
9	Relatório de testes	x	
10	Registro de rastreabilidade	x	
11	Registro de verificação		x
12	Registro de validação		x

Fonte: ABNT NBR ISO/IEC 29110-4 (2012)

3.4 Considerações do Capítulo

Neste capítulo, foram apresentados 3 modelos de referência para melhoria do processo de desenvolvimento de software. Dois deles, o CMMI-DEV e o MPS.BR-SW são modelos de maturidade e capacidade. Em ambos modelos, as melhorias no processo estão associadas a níveis, ou seja, os processos implementados dependem do nível adotado pela companhia. Já o terceiro modelo apresentado, é a norma ISO/IEC 29110-4, criada para ser adotada por pequenas empresas. Neste caso, a norma define os processos que serão melhorados, sendo eles os processos de Gerenciamento de Projetos e Implementação de Software, cabendo à empresa definir a metodologia de implantação.

Devido a publicação da norma ISO/IEC 29110-4 ainda ser relativamente recente, seu processo de adoção ainda é pouco conhecido no Brasil. O modelo mais conhecido atualmente é o CMMI-DEV, que possui reconhecimento internacional, porém trata-se de um modelo criado para ser aplicado em grandes projetos do Departamento de Defesa (DoD) norte-americano, e mais adequado às grandes companhias. Na opinião de Pinheiro (2010), a adoção do modelo CMMI-DEV implica em um grande custo e tempo a ser investido. Além disso, quanto maior o nível de maturidade, maior a quantidade de recursos humanos necessários para o cumprimento dos processos, contrastando com a realidade das pequenas empresas brasileiras.

O MPS.BR-SW é um modelo nacional e publicado em 2010. Sua adoção em VSEs é adequada somente até os níveis G e F. Após esses níveis iniciais, o modelo é tão complexo quanto o CMMI-DEV e possui somente reconhecimento nacional.

Apesar de a norma ISO/IEC 29110-4 possuir um escopo mais enxuto em relação aos modelos CMMI-DEV e MPS.BR-SW, existe um alto grau de equivalência entre os 3 modelos. De acordo com Posca e Hiramã (2015), foram identificados índices de equivalência de, respectivamente, 67% e 78% no MPS.BR-SW-SW, 67% e 81% no CMMI-DEV, em relação aos processos de Gerenciamento de Projetos e Implementação de Software da norma.

Ainda de acordo com Posca e Hiramã (2015), a norma possui elementos que não estão presentes nos modelos CMMI-DEV e MPS.BR-SW, como a Declaração de Trabalho e a atividade de Encerramento do Projeto.

Abaixo estão relacionados os principais pontos positivos e negativos em relação aos modelos CMMI-DEV e MPS.BR-SW (Pinheiro, 2010):

Modelo CMMI-DEV:

- Pontos positivos
 - Visibilidade reconhecida internacionalmente;
 - Fornece cobertura detalhada do ciclo de vida do produto;
- Pontos Negativos
 - Custo e tempo para implantação;

Modelo MPS.BR-SW-SW:

- Pontos positivos
 - Compatível com CMMI-DEV;
 - Possui sete níveis de maturidade, onde a implantação é mais gradual e adequada a pequenas e médias empresas;
- Pontos negativos
 - Não apresenta competitividade internacionalmente com o CMMI-DEV;

Com base nos motivos aqui expostos, a norma ISO/IEC 29110-4 será utilizada como base do estudo de caso. Será definido um processo aderente à norma, e utilizado em um projeto real de uma pequena empresa.

4. ESTUDO DE CASO: APLICAÇÃO DAS BOAS PRÁTICAS EM UMA PEQUENA EMPRESA

Como foi observado capítulos anteriores, o uso de boas práticas para o desenvolvimento de software é essencial para que o produto final seja entregue com qualidade e no prazo. Existem diversos modelos de boas práticas aplicados aos processos de desenvolvimento, porém nem todos são viáveis para as pequenas empresas. Após elencar os principais pontos positivos e negativos dos modelos CMMI-DEV, MPS.BR-SW e a norma ISO/IEC 29110-4, esta última foi escolhida para o estudo de caso, onde ela será implementada em uma pequena empresa brasileira.

O objetivo é que a metodologia apresentada a seguir possa ajudar outras pequenas empresas na busca pela melhoria de seus processos de desenvolvimento de software.

4.1 A Empresa

A empresa estudada é nacional e foi fundada em 2005 na cidade de São Paulo, SP. Sua principal atividade econômica é prestar serviços de rastreamento de mercadorias. Por motivos de confidencialidade, a empresa será chamada pelo nome fictício de LogSoft.

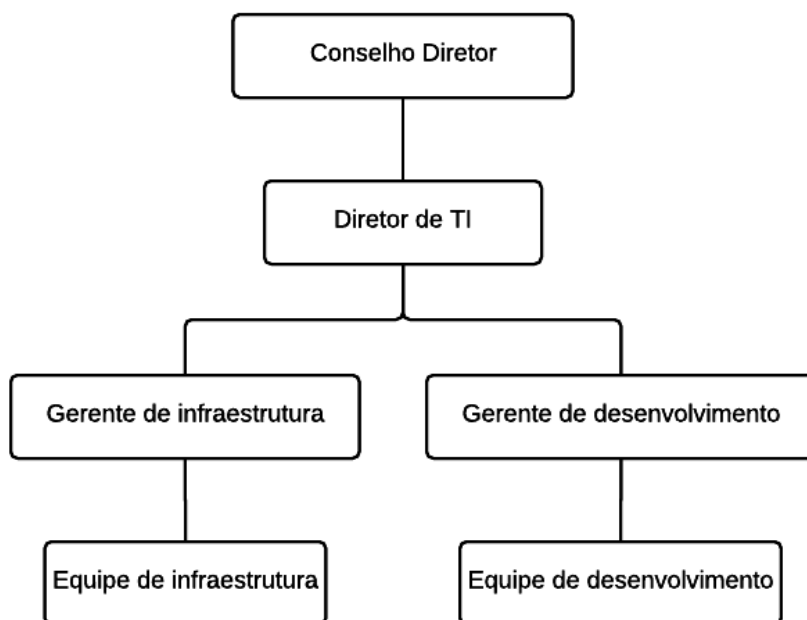
O serviço de rastreamento é feito através do acompanhamento dos veículos que fazem o transporte de mercadorias via rastreadores GPS. O principal objetivo deste serviço é inibir o roubo de mercadorias. Empresas atuantes neste nicho de mercado são denominadas Gerenciadoras de Risco. A necessidade de empresas gerenciadoras de risco é fruto dos problemas de segurança pública existentes no Brasil. O serviço prestado pelas empresas deste nicho é uma exigência das seguradoras. Em muitos casos, uma carga está coberta pela apólice de seguro somente quando se comprova a existência de uma empresa gerenciadora de risco acompanhando suas operações de transporte.

A atividade de se acompanhar o transporte de mercadorias em um país tão grande como o Brasil é possível graças aos avanços na tecnologia de rastreadores veiculares baseados em GPS e na possibilidade da comunicação via satélite entre o veículo e a central de rastreamento. A possibilidade de se obter em tempo real a posição do

veículo, representada como um conjunto de coordenadas latitude e longitude, possibilita a criação de sistemas de acompanhamento logístico. Entre as possibilidades, é possível calcular a velocidade média de um veículo medindo seu deslocamento entre dois conjuntos de coordenadas. Com essa informação, a previsão de chegada do veículo pode ser estimada. Dessa forma, a empresa oferece serviços de gerenciamento de riscos de transporte e logística.

A figura 07 mostra como o departamento de desenvolvimento de software se situa dentro da empresa.

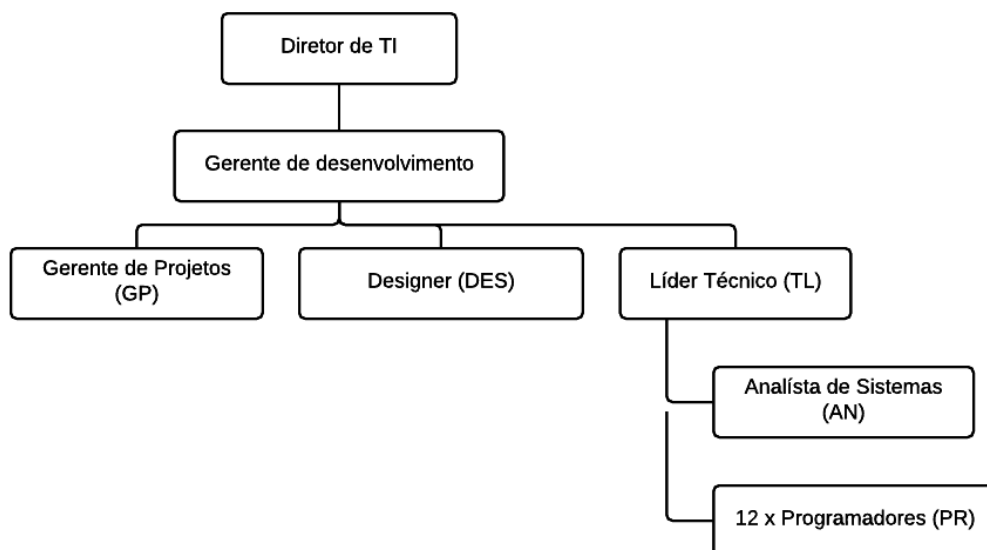
Figura 7 - Organograma da área de TI



Fonte: Autor

A figura 8 descreve o organograma interno da equipe, ou seja, a estrutura hierárquica entre os papéis dentro do departamento de desenvolvimento de software.

Figura 8 - Organograma do departamento de desenvolvimento de software



Fonte: Autor

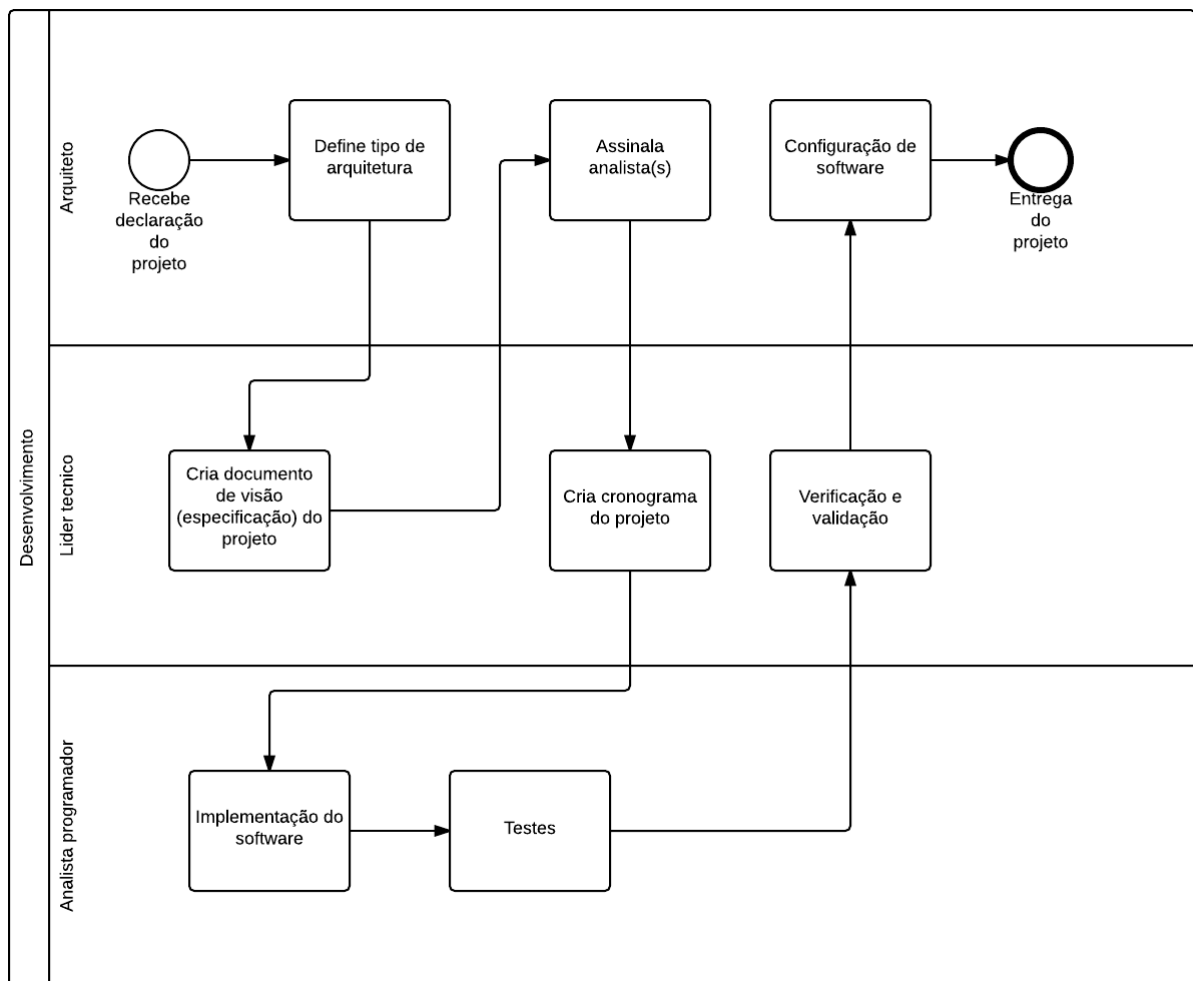
Cada um dos papéis presentes no organograma do departamento possui seu respectivo conjunto de responsabilidades. A tabela 13 descreve as responsabilidades dos papéis com participação direta no processo de desenvolvimento.

Tabela 13 - Matriz de papéis e responsabilidades

Papel	Responsabilidades
Gerente de Projetos	1. Planejar o projeto; 2. Acompanhar o andamento do projeto; 3. Controlar desvios e mudanças em relação ao planejado inicialmente; 4. Alinhar as expectativas do cliente em relação ao projeto;
Arquiteto de Software	1. Criar o modelo arquitetural do software; 2. Modelar e projetar componentes de software; 3. Planejar e realizar os testes de integração;
Líder Técnico	1. Criar e atualizar especificação técnica; 2. Criar e atualizar e acompanhar o cronograma do projeto; 3. Apoiar e tirar dúvidas técnicas da equipe;
Analista	1. Coletar requisitos funcionais e não funcionais; 2. Validar e verificar o software;
Programador	1. Desenvolver softwares de acordo com as especificações; 2. Realizar testes unitários e de integração; 3. Persistir o código fonte no repositório do projeto; 4. Informar andamento do projeto;
Cliente	1. Definir e aprovar os requisitos do software 2. Definir e aprovar as solicitações de mudança; 3. Realizar o aceite do software;

A equipe de desenvolvimento não possui um processo formal e descrito de desenvolvimento, porém segue um conjunto mínimo de atividades necessárias para que o software seja construído. As atividades apresentadas no fluxo abaixo são executadas em todos os projetos, porém nem sempre obedecem esta ordem. Além disto, são executadas de forma diferente em cada projeto.

Figura 9 - O atual processo de desenvolvimento



Fonte: Autor

4.2 Situação Atual

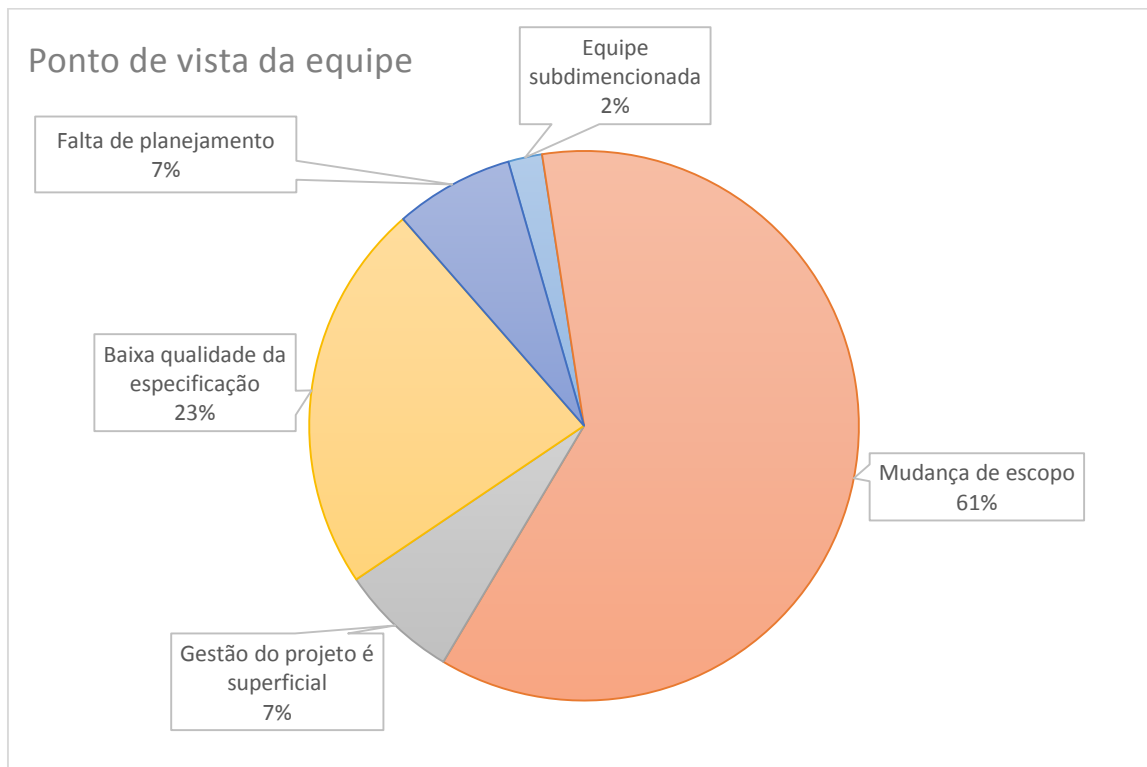
A empresa estudada apresenta problemas com a qualidade das entregas e gestão dos projetos de software. Não são raros os casos onde o prazo para entrega do software não é cumprido. Além disso, o escopo do software é frequentemente mal definido, acarretando retrabalho para a equipe de desenvolvimento e em um software que não atende às expectativas do cliente. Portanto, a situação atual acarreta prejuízos à empresa e aos seus clientes.

Com o intuito de se identificar os principais problemas enfrentados, foi questionado à equipe de desenvolvimento (12 pessoas) e diretoria (4 pessoas), quais são os principais fatores que influenciam negativamente os projetos da empresa. Devido ao pouco tempo cedido para a realização deste levantamento na empresa, cada pessoa poderia escolher somente 3 opções das descritas abaixo.

- a. Equipe subdimensionada
- b. Equipe superdimensionada
- c. Aceitar mudança de escopo durante o desenvolvimento
- d. Falta de melhores ferramentas
- e. Gestão do projeto é superficial
- f. Baixa qualidade da especificação e definição dos requisitos
- g. Falta de planejamento para a condução dos projetos

Na visão da equipe de desenvolvimento os problemas são:

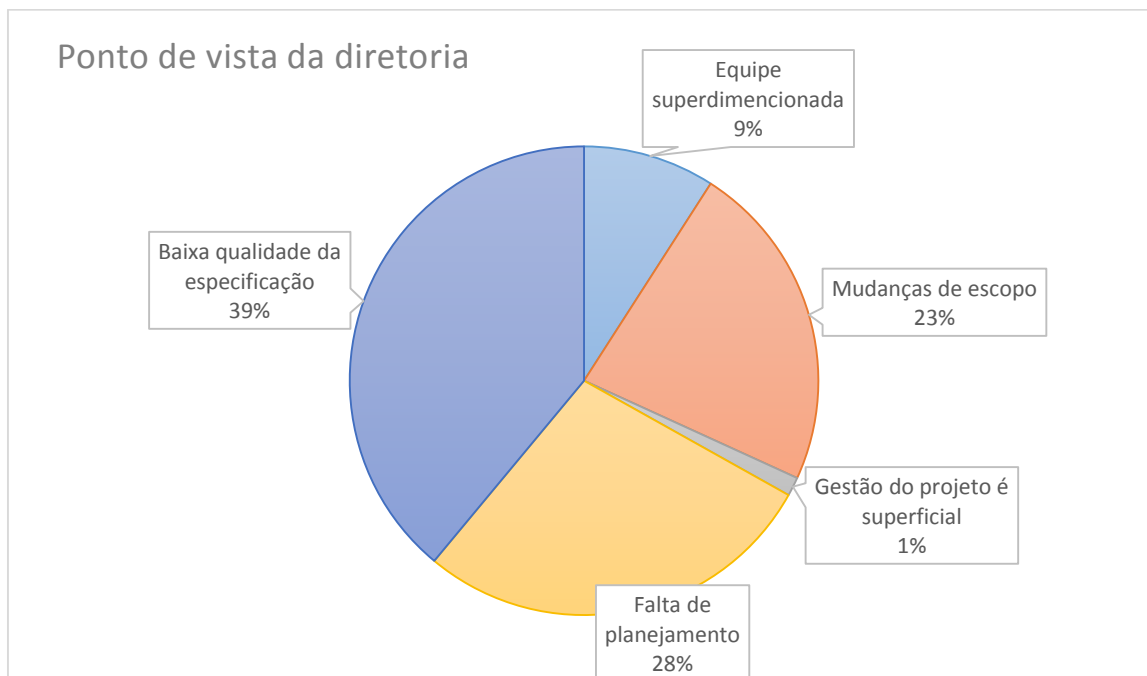
Figura 10 - Fraquezas sob a ótica da equipe



Fonte: Autor

Na visão dos sócios diretores:

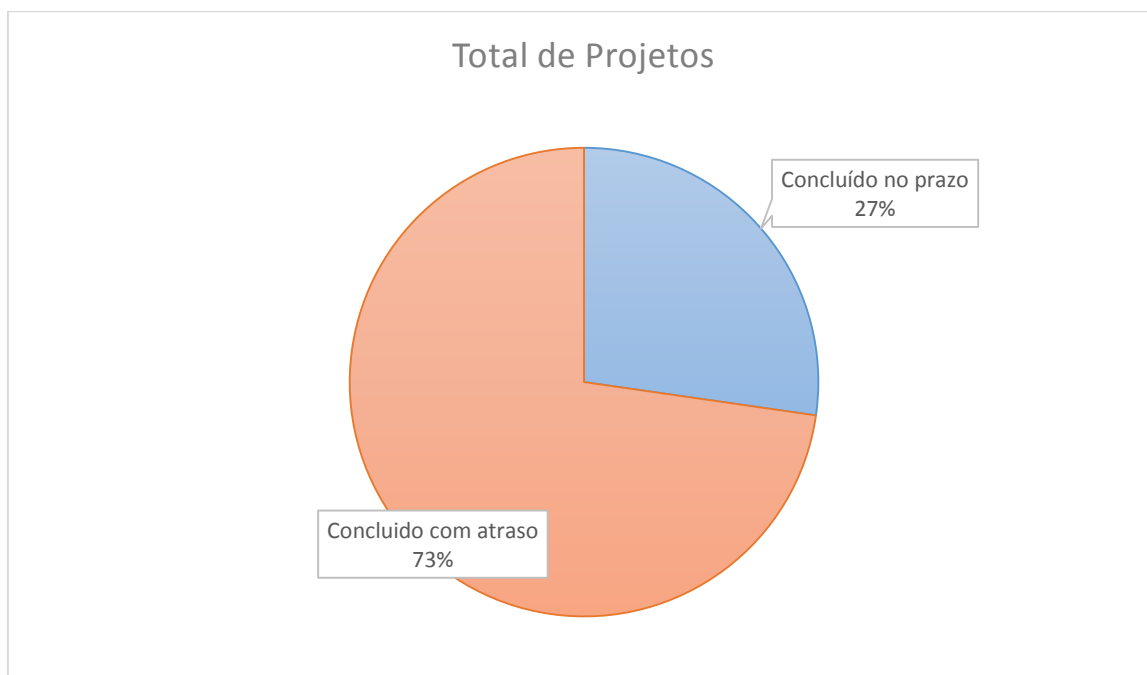
Figura 11 - Fraquezas, sob a ótica da diretoria



Fonte: Autor

Além da enquete, foi realizada uma análise do histórico dos projetos entre o período de janeiro de 2014 e fevereiro de 2015. Este período contou com um total de oito projetos atrasados dentro de um total de quinze projetos. A partir desta análise, obtivemos os seguintes dados.

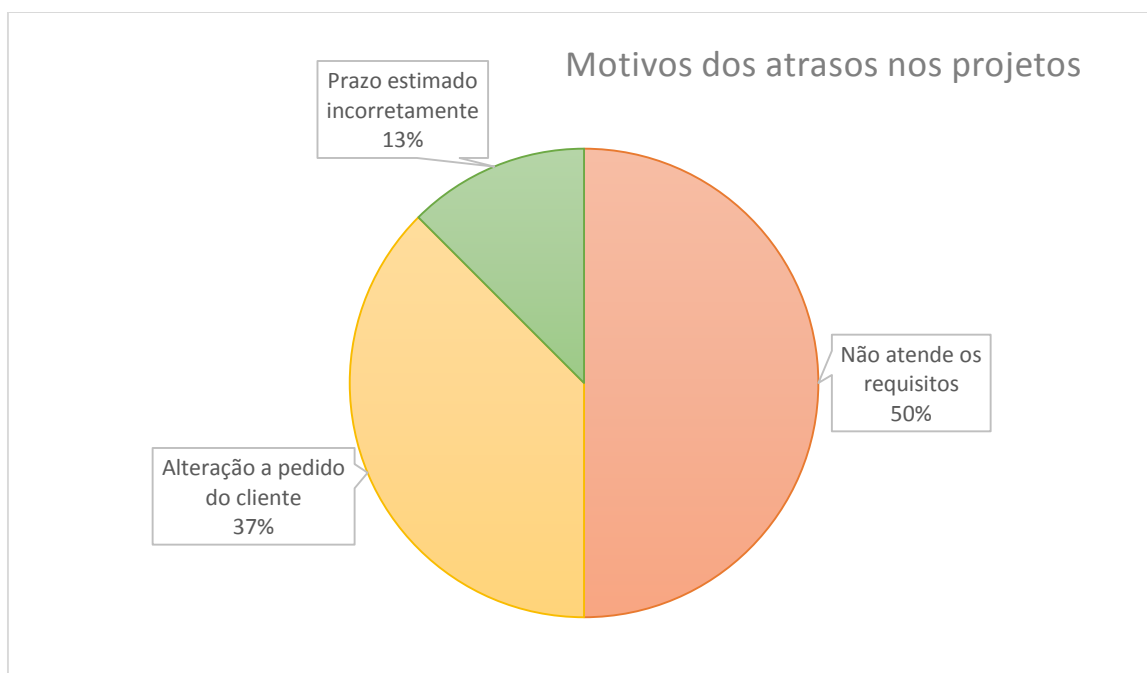
Figura 12 - Status dos projetos



Fonte: Autor

A figura 13 exibe o gráfico dos motivos dos projetos atrasados.

Figura 13 - Projetos atrasados



Fonte: Autor

Através dos dados coletados na empresa e expostos anteriormente, as seguintes afirmações podem ser feitas:

- Prazos não são atendidos, prioritariamente, porque o software não atende aos requisitos, gerando diversos pedidos de correções por parte do cliente;
- Requisitos não são validados e formalmente aprovados pelo cliente;
- Software não é testado nem validado formalmente pelo cliente, acarretando em um produto final com baixa aderência aos requisitos;
- O escopo do projeto é mal definido;

A tabela 14 foi criada a partir das principais falhas apresentadas atualmente pela empresa.

Tabela 14 - Principais falhas do processo de desenvolvimento atual

Problema	Motivo	Afeta
Prazos não são atendidos	Falta de planejamento, análise de riscos e disponibilidade de recursos; Mudanças do escopo do projeto;	Clientes
Escopo mal definido	A empresa não possui um modelo padrão de escopo e não exige aprovação do escopo levantado pelo cliente	Empresa, equipe desenvolvimento
O processo não é seguido	A empresa não divulga o processo nem cobra/fiscaliza seu cumprimento	Equipe desenvolvimento
O software não é testado nem validado	Não existe um processo definido em que se atesta se o que foi produzido atende as necessidades do cliente	Clientes

O software não atende aos requisitos do cliente	Falta de processo de verificação e validação; O software é entregue sem estar pronto somente para cumprir o prazo acordado.	Clientes
Cliente fez mudanças no escopo do projeto a qualquer momento	A empresa aceita que os clientes façam mudanças no escopo do projeto a qualquer momento, sem que qualquer penalização seja aplicada	Equipe desenvolvimento, Empresa
Especificação técnica não é feita, ou é superficial	Pouco tempo para que o projeto (<i>design</i>) de software seja feito com qualidade	Equipe desenvolvimento
Somente um membro da equipe conhece as regras de negócio	Falta de documentação; Processo não é cumprido; Centralização das responsabilidades em somente um membro da equipe;	Equipe desenvolvimento

Fonte: Autor

4.3 Metodologia para Melhoria do Processo

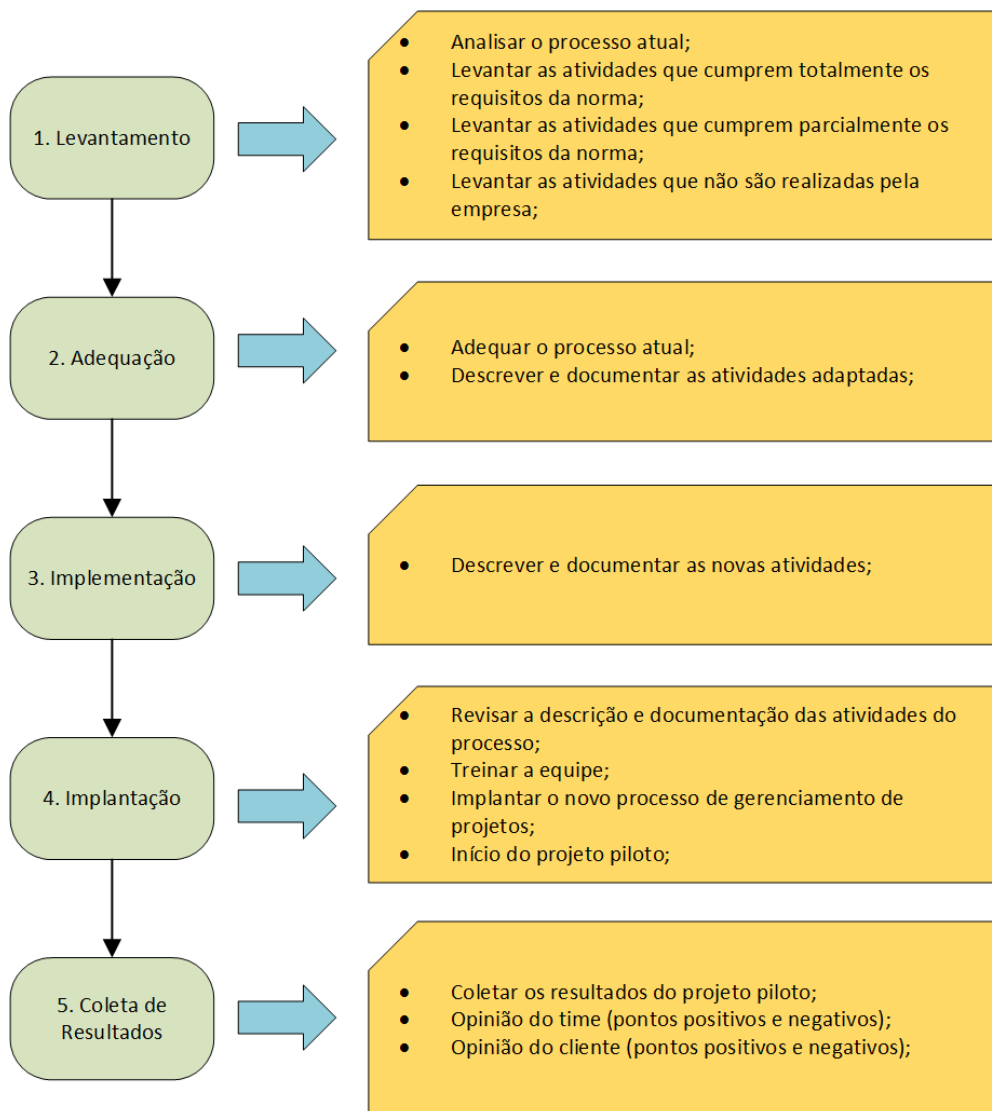
Com base nas motivações descritas anteriormente, o objetivo é contribuir para a melhoria da competitividade de pequenas empresas, aplicando as boas práticas da engenharia de software a fim de se melhorar o processo. No caso da empresa LogSoft, será apresentada a implantação da norma ISO/IEC 29110-4 de forma a minimizar os problemas encontrados atualmente, que hoje interferem negativamente na qualidade de seu software e em sua competitividade. Este estudo de caso se limita a implementação do processo de Gerenciamento de Projetos (PM).

Através da adoção das boas práticas propostas pela norma, se pretende que a empresa tenha um processo bem definido, com um maior aporte em relação a gestão dos seus projetos de software. Além disso, pretende-se que o conhecimento a respeito da norma (que ainda é pouco conhecida) e a metodologia de implantação aqui apresentada possam contribuir para outras pequenas empresas com os mesmos problemas da LogSoft.

Durante a análise do departamento e mapeamento de suas principais falhas, a principal carência está relacionada a forma como o projeto é gerenciado. A falta de um plano que norteie a equipe durante as fases do desenvolvimento de software e de uma rotina de acompanhamento do andamento do projeto, e principalmente, a falta do aceite do cliente em relação ao escopo previamente acordado entre ambas as partes são as características que mais contribuíam para a relativa baixa qualidade do software produzido na empresa por sua equipe de desenvolvimento.

O departamento de software da empresa vem trabalhando de forma pouco padronizada há aproximadamente 2 anos. A adoção de um processo baseado na norma ISO/IEC 29110-4 envolve uma grande mudança de cultura para todos os membros da equipe. Uma metodologia de implantação da norma foi criada, visando diminuir possíveis impactos negativos na equipe. A figura 14 descreve os passos percorridos até a criação de um processo aderente à norma.

Figura 14 - Metodologia de implantação do processo PM



Fonte: Autor

A etapa de Levantamento compreende a análise do processo atual visando encontrar pontos onde a atividade executada atenda aos requisitos da norma, total ou parcialmente. Desta forma, estas atividades já existentes podem servir de alicerce para a implementação da norma. Adaptar uma tarefa ou atividade já existente pode facilitar a assimilação do novo processo pela equipe, e diminuir os impactos da mudança.

Os seguintes critérios foram usados para classificar as atividades do processo atual:

- Atende totalmente: a atividade executada atualmente atende totalmente os requisitos da norma;
- Atende parcialmente: a atividade executada atualmente possui semelhança aos requisitos da norma, porém ajustes serão necessários para que estes requisitos sejam atendidos 100%;
- Não atende: não foram encontrados paralelos entre a atividade e os requisitos da norma, neste caso será criada uma atividade ou tarefa que atenda aos requisitos da norma;

A tabela 15 apresenta a situação do processo atual, em relação aos 7 objetivos do processo de Gestão de Projetos da norma ISO/IEC 29110-4.

Tabela 15 - Atual situação em relação aos objetivos do processo PM

Objetivos	Cumprimento a norma
PM.O1 Plano do projeto	Não atende
PM.O2 Registro do progresso do projeto	Atende parcialmente
PM.O3 Gestão de mudança	Não atende
PM.O4 Reuniões de revisão	Atende parcialmente
PM.O5 Identificação de riscos	Não atende
PM.O6 Estratégia de controle de versão	Atende totalmente
PM.O7 Garantia da qualidade	Não atende

Fonte: Autor

A situação do processo atual, em relação às 4 atividades do processo PM da norma ISO/IEC 29110-4 estão listadas abaixo.

Tabela 16 - Atual situação em relação às atividades do processo PM

Atividades	Cumprimento a norma
PM.1 Planejamento do projeto	Atende parcialmente
PM.2 Execução do plano do projeto	Atende parcialmente
PM.3 Avaliação e controle do projeto	Não atende
PM.4 Encerramento do projeto	Não atende

Fonte: Autor

A seguir está listada a situação do processo atual, em relação aos produtos de trabalho do processo PM.

Tabela 17 - Atual situação em relação aos produtos de trabalho do processo PM

Produto de trabalho	Cumprimento a norma
Plano do projeto	Não atende
Registro de aceite	Não atende
Solicitação de mudança	Não atende
Registro de reunião	Atende parcialmente
Guia de operação do produto/manual	Não atende
Registro de status do progresso	Atende parcialmente
Repositório do projeto	Atende totalmente
Backup do repositório do projeto	Atende totalmente

Fonte: Autor

Com base no levantamento realizado, foi possível usar o processo atual como ponto de partida para a criação do novo processo de desenvolvimento, aderente à norma ISO/IEC 29110-4.

Durante a próxima etapa, Adequação, todos os ajustes necessários às atividades atuais foram feitos e documentados. A terceira etapa, Implementação, representou a criação das atividades, tarefas e produtos de trabalho. Foram criados *Templates* para os novos produtos de trabalho.

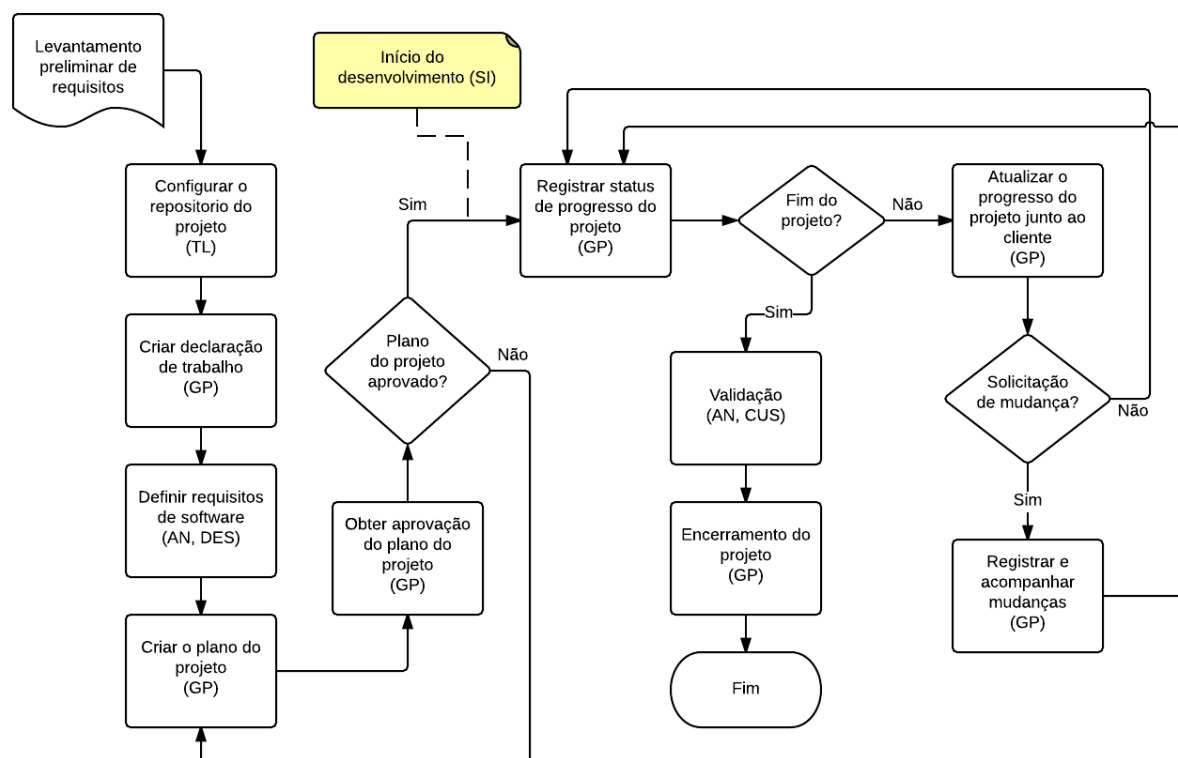
Após as etapas de adequação do processo atual e implementação das novas atividades e tarefas para atendimento aos requisitos da norma, o novo processo foi divulgado à equipe. A documentação do processo foi criada (figura 15), abrangendo os seguintes pontos:

- Fluxo do processo;
- Manual, com descrição de cada uma das atividades;
- Diretório de rede contendo todos os templates e modelos usados para confecção dos produtos de trabalho;
- Definido o coordenador da equipe como responsável pelo processo. Durante a implantação do novo processo, o coordenador é o principal facilitador (para tirar dúvidas) e fiscalizador (garantir cumprimento) do novo processo.

4.4 O Novo Processo de Desenvolvimento

Conforme escopo deste trabalho, imagem 15 descreve o fluxo de atividades do novo processo de gerenciamento de projeto, desenhado de acordo com a norma ISO/IEC 29110-4.

Figura 15 - Fluxo de atividades do novo processo



Fonte: Autor

A tabela 18 lista os produtos de trabalho do novo processo, incluindo os 8 produtos de trabalho requeridos pelo processo PM, acrescido de um produto de trabalho do processo SI.

Tabela 18 - Produtos de trabalho

	Produtos de trabalho	Processo
1	Plano do projeto	PM
2	Registro de aceite	PM
3	Solicitação de mudança	PM
4	Repositório do projeto	PM
5	Backup do repositório do projeto	PM
6	Registro de reunião	PM
7	Registro de correção	PM
8	Registro de status do projeto	PM
9	Especificação dos requisitos de software	SI
10	Registro de validação	SI
11	Declaração de trabalho	-

Fonte: Autor

Conforme dito anteriormente, o foco deste estudo de caso é a implementação do processo PM, definido pela norma ISO/IEC 29110-4, em uma VSE. Porém, como pode ser notado pela tabela 18, dois produtos de trabalho do processo SI foram incluídos (item 9 e 10).

A inclusão do item Especificação dos requisitos de software, e suas respectivas atividades, tem o objetivo de suprir uma das principais fraquezas das pequenas empresas em geral, que é a falta de uma especificação de software bem definida, aprovada pelo cliente.

Já a inclusão do item Registro de validação visa introduzir suprir as fraquezas relacionadas à qualidade do software desenvolvido. O Registro de validação fornece rastreabilidade entre os requisitos do produto e o software desenvolvido.

O item 11 (Declaração de Trabalho) é um pré-requisito do projeto, porém não pertence aos processos PM e SI. A Declaração de Trabalho é o escopo do produto e serve de base para o Plano do Projeto e Especificação dos Requisitos.

Os papéis usados na descrição das atividades do processo estão definidos na tabela 13. Os próximos tópicos se dedicam a explicar cada etapa do novo processo.

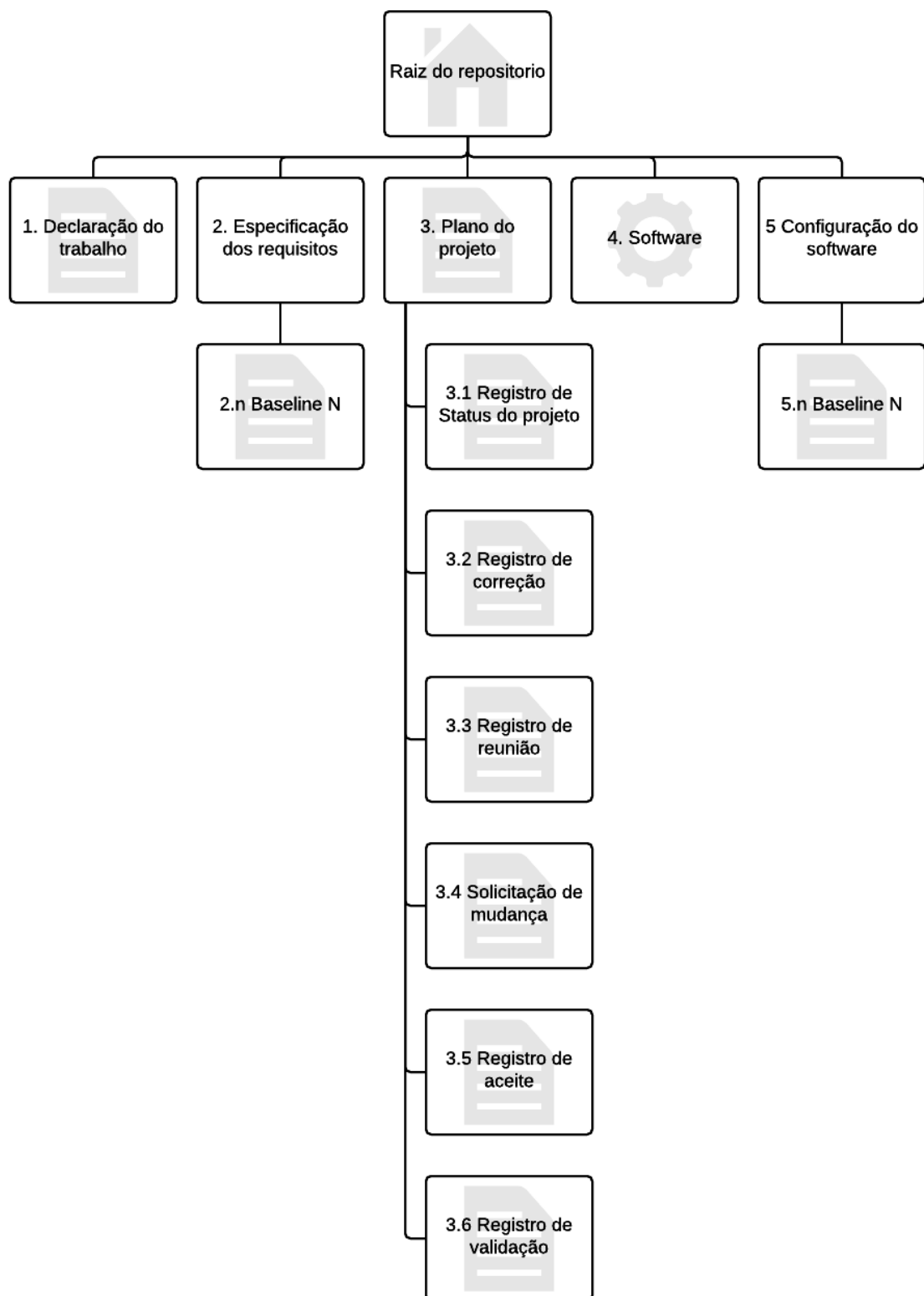
4.4.1 Configurar o Repositório do Projeto

O repositório do projeto é o local onde todas as informações referentes ao projeto devem ser armazenadas. O repositório deve permanecer acessível para a equipe do projeto. O repositório do projeto deve obedecer a estratégia de controle de versão definida pelo departamento de desenvolvimento de software. Neste caso, optou-se por desenvolver uma estratégia de controle de versão universal, aplicada a todos os projetos.

Foi utilizada uma ferramenta *opensource* e *freeware* (SVN) destinada ao controle de versão de arquivos. Todos os artefatos do projeto e código fonte estão armazenados no servidor e cada versão ou alteração dos arquivos é persistida. Caso um erro seja cometido, a alteração pode ser facilmente revertida para a versão anterior/inicial. O servidor Visual SVN possui uma rotina de backup diário. Cada backup é armazenado por 30 dias.

A figura 16 apresenta a estrutura do repositório do projeto.

Figura 16 - Estrutura do repositório do projeto



Produtos de Entrada: Estratégia de controle de versão

Produtos de Saída: Repositório do projeto, Backup do repositório do projeto

Papel: TL (Líder Técnico)

Atividade: PM.1

Objetivo: PMO.6

4.4.2 Criar a Declaração de Trabalho

A declaração de trabalho é o documento que define o escopo do produto de software, em outras palavras, ela define o propósito do projeto e todos os entregáveis do projeto que devem ser entregues ao cliente.

O escopo do produto define as fronteiras e limites do software, ou seja, tudo aquilo que o software deve realizar (por exemplo: cálculos, algoritmos, exibição de dados) e aquilo qual o software não foi concebido para realizar (limitações legais, falta de viabilidade técnica, limitações de software e/ou hardware).

A Declaração de Trabalho não contém informações de preço/custo do projeto, portanto ela não é equivalente a um contrato. Porém, a mesma é anexada ao contrato firmado entre cliente e empresa produtora de software e será usada como referência durante todo o processo de desenvolvimento e entrega do produto de software.

O Gerente de Projetos (GP) é o responsável por confeccionar a Declaração de Trabalho. Ele deve utilizar como base o Levantamento Preliminar de Requisitos. Este documento é criado pelo Analista (AN) ainda durante a fase de negociação comercial entre cliente e a empresa produtora de software. Por se tratar de uma atividade de suporte ao departamento comercial, não será abordado neste trabalho o processo de criação do Levantamento Preliminar de Requisitos, porém o mesmo trata-se de uma lista de requisitos funcionais levantados durante análise das necessidades do cliente. Muitas vezes somente a Declaração Preliminar de Requisitos não satisfaz totalmente o nível de informações e detalhes requeridos pela Declaração do Trabalho. Neste caso, se faz necessário o envolvimento do Analista (AN) para uma nova rodada de levantamento de requisitos e/ou detalhamento dos requisitos já levantados. Este

processo pode ser feito via entrevistas com os usuários do software, questionários, reuniões com o cliente sendo presenciais ou via teleconferência.

A lista a seguir relaciona os tópicos abordados pela Declaração de Trabalho.

- **Introdução:** contém uma breve descrição das empresas envolvidas no projeto, o propósito do projeto, definições, nomenclaturas e abreviações;
- **Stakeholders:** descrição dos principais *stakeholders* do projeto e suas respectivas responsabilidades;
- **Escopo do Trabalho:** breve descrição do processo de desenvolvimento, etapas e fases do projeto, metodologia utilizada, descrição do local onde o trabalho será realizado, possíveis itens que não fazem parte do projeto;
- **Descrição do produto:** principais objetivos e resultados esperados com a adoção do produto, descrição dos requisitos do software, definição da arquitetura da solução, pressupostos e dependências, integração entre sistemas e demais descrições relacionadas ao produto;
- **Riscos do projeto:** riscos identificados até o início do projeto;
- **Condições de entrega:** lista de condições que devem ser atendidas para que o projeto seja considerado concluído;
- **Manutenção e suporte:** detalhamento da estratégia de suporte e processo de manutenção (correção de falhas) do produto;

Produtos de Entrada: Análise preliminar de requisitos

Produtos de Saída: Declaração de Trabalho

Papel: Gerente de projetos (GP) e Analista (AN)

Atividade: PM.1

Objetivo: PMO.1, PMO.5

4.4.3 Definir Requisitos de Software

Conforme visto anteriormente, os requisitos de software estabelecem quais necessidades dos *stakeholders* devem ser atendidas pelo software (capítulo 2).

A lista a seguir relaciona os tópicos abordados pela tarefa Definir Requisitos de Software.

- **Introdução:** fornece um resumo do produto e seu propósito, deve conter um breve resumo do escopo, definições, terminologia e referências externas;
- **Descrição do ambiente:** deve descrever de forma geral os fatores que afetam o produto e seus requisitos. Ex: características dos usuários, dependências externas, legislação, nicho de indústria a que se destina o produto, infraestrutura de TI;
- **Requisitos funcionais:** esta seção descreve os requisitos funcionais e não funcionais do software. A descrição deve conter um nível de detalhamento suficiente para que o Designer (DES) possa arquitetar o software e para que a equipe de testes possa verificar se o software satisfaz os requisitos;
- **Requisitos de usabilidade:** Descrever requisitos não funcionais de usabilidade, ou seja, requisitos que definem características de usabilidade (interface homem-sistema) do software;
- **Requisitos de confiabilidade:** seção destinada a descrever requisitos não funcionais de confiabilidade, ou seja, requisitos que definem características de confiabilidade do software (por exemplo: sistema deve permanecer online 24x7);
- **Requisitos de desempenho:** sessão destinada a descrever requisitos não funcionais de desempenho (por exemplo: sistema deve realizar cálculo em menos de 5 segundos);

Produtos de Entrada: Declaração de Trabalho

Produtos de Saída: Especificação dos requisitos de software, Declaração de Trabalho (Atualizada)

Papel: Analista (AN) e Projetista (DIS)

Atividade: SI.2

Objetivo: SI.O2, SI.O6 e SI.O7

4.4.4 Criar Plano do Projeto

O Plano do Projeto pode ser visto como o principal produto do processo de Gerenciamento de Projetos (PM) pois ele deverá servir para a equipe de trabalho como um guia durante todas as etapas do projeto, desde a definição do escopo do produto, até a entrega do software finalizado.

O objetivo do Plano do Projeto é apresentar a forma como os processos e atividades do projeto serão executados para garantir a conclusão bem-sucedida do projeto, bem como a qualidade dos produtos entregáveis.

A lista a seguir relaciona os tópicos abordados pela tarefa Criar Plano do Projeto.

- **Introdução:** objetivos do documento, público a quem ele se destina, considerações iniciais;
- **Descrição do produto:** descrição do produto, relação de requisitos atualizada com base na declaração de trabalho;
- **Escopo do projeto:** escopo do projeto, ou seja, tudo que está incluído no projeto;
- **Objetivos do projeto:** principais objetivos, benefícios, responsabilidades;
- **Entregáveis:** lista de produtos/artefatos a serem entregues ao cliente
- **Instruções de entrega:** descreve como o produto será entregue (definições como onde será hospedado, entregas em etapas, entrega da documentação ao cliente, entrega do código fonte);
- **Tarefas:** lista de tarefas que devem ser concluídas para que o produto seja construído e entregue, incluindo as tarefas relacionadas ao ciclo de vida do desenvolvimento de software, como análise, design, implementação e testes. As tarefas podem ser representadas como uma Estrutura Analítica do Projeto (EAP);

- **Relacionamento e dependência entre as tarefas:** lista de tarefas e subtarefas de forma a se demonstrar seu relacionamento e evidenciar quais etapas podem ser feitas de forma paralela ou não;
- **Duração estimada das tarefas:** estimativa do esforço em horas necessário para o cumprimento de determinada tarefa;
- **Recursos:** pessoas, materiais, normas, equipamentos e ferramentas, bem como o calendário de quando os recursos são necessários;
- **Composição da equipe de trabalho:** descrição da equipe (nome, papel, responsabilidades);
- **Estimativa de custo e esforço:** com base na estimativa de esforço em horas, deve se estimar o custo do projeto;
- **Identificação dos riscos:** principais riscos para a realização do projeto e possíveis formas de mitigar tais riscos (ex: risco devido ao baixo nível técnico da equipe, uma solução seria proporcionar treinamento técnico para seus membros);
- **Estratégia de controle de versão:** repositórios do projeto, regras de controle de versão, regras de nomenclatura das versões do software;
- **Acompanhamento do projeto:** descreve como o Gerente de Projetos deve acompanhar o projeto, a frequência com que o Registro de Status do Projeto deve ser atualizado, frequência das reuniões de alinhamento junto ao cliente, formato das reuniões (presenciais, teleconferência, videoconferência);

Produtos de Entrada: Declaração do Trabalho, Especificação dos requisitos de software

Produtos de Saída: Plano do Projeto

Papel: Gerente de projetos (PM), Líder Técnico (TL)

Atividade: PM.1

Objetivo: PMO.1, PMO.7

4.4.5 Aprovação do Plano do Projeto

O objetivo desta tarefa é verificar se todos os elementos do Plano do Projeto são viáveis e consistentes. Os resultados encontrados são documentados como

“resultados da verificação” e correções são feitas até que o documento seja aprovado pelo PM e pelo cliente.

Produtos de Entrada: Plano do Projeto

Produtos de Saída: Registro de aceite, Plano do Projeto aprovado pelo Gerente de Projetos (PM) e cliente (CUS)

Papel: Gerente de projetos (PM), Cliente (CUS)

Atividade: PM1

Objetivo: PMO.1, PMO.5, PMO.6, PMO.7

4.4.6 Registrar Status do Projeto

O gerente de projetos monitora a execução do projeto de acordo com o Plano do Projeto e grava dados reais sobre o status, andamento, tarefas realizadas e tarefas pendentes no documento chamado “Registro do progresso do projeto”. Este documento serve como registro histórico das etapas do projeto, principais problemas enfrentados durante a construção do software e possíveis atrasos.

A frequência com que o status do projeto é registrado deve ser definida no Plano do Projeto, porém caso o projeto apresente dificuldades é recomendável que o registro de status seja feito com intervalo menor.

A lista a seguir relaciona os tópicos abordados pela tarefa Registrar o Progresso do Projeto.

- **Introdução:** objetivos do documento, a quem se destina, considerações iniciais;
- **Data do status:** data em que o reporte do status do projeto foi realizada;
- **Equipe do projeto:** nome das pessoas responsáveis por informar o status do projeto e de suas respectivas tarefas;
- **Tarefas:** Registrar o status das tarefas atuais em relação as tarefas planejadas;
- **Objetivos:** Registrar os resultados atuais em relação aos objetivos planejados;
- **Alocação de Recursos:** Registrar o status de alocação de recursos em relação ao planejamento de alocação de recursos;

- **Custos:** Registrar o custo atual do projeto em relação ao planejamento/estimativa de custos;
- **Cronograma:** Registrar progresso do projeto em relação ao cronograma/prazo estimados;
- **Riscos:** Registrar o status dos riscos atuais em relação aos riscos identificados previamente e registrar novos riscos, se identificados;
- **Deshvios:** Registrar desvios das tarefas em relação ao planejado, informando o motivo dos desvios;

Produtos de Entrada: Plano do projeto

Produtos de Saída: Registro de status do projeto

Papel: Gerente de projetos

Atividade: PM.2

Objetivo: PM.O2, PM.O3, PM.O4, PM.O5, PM.O7

4.4.7 Atualizar o Progresso do Projeto Junto ao Cliente

De acordo com a norma ISO/IEC 29110-4, reportar o status do projeto junto ao cliente é fundamental para alinhar as expectativas, tirar dúvidas, obter *feedback* e, caso necessário, anteceder a introdução de mudanças durante o processo de desenvolvimento.

A execução desta tarefa, cuja responsabilidade é do gerente de projetos, utiliza os dados coletados na tarefa anterior, como a coleta de status junto a equipe do projeto, identificação das tarefas do projeto já concluídas, identificação das tarefas ainda pendentes, situação do projeto em relação ao prazo e custo, dúvidas a respeito dos requisitos e regras de negócio (caso existam).

Durante a reunião com o cliente, mudanças podem ser introduzidas. Caso existam, o gerente de projetos deverá proceder com a abertura de uma solicitação de mudança (descrita no tópico Registrar e Acompanhar mudanças).

Produtos de Entrada: Plano do Projeto, Registro de status do projeto

Produtos de Saída: Registro de reunião

Papel: Gerente de projetos

Atividade: PM.3

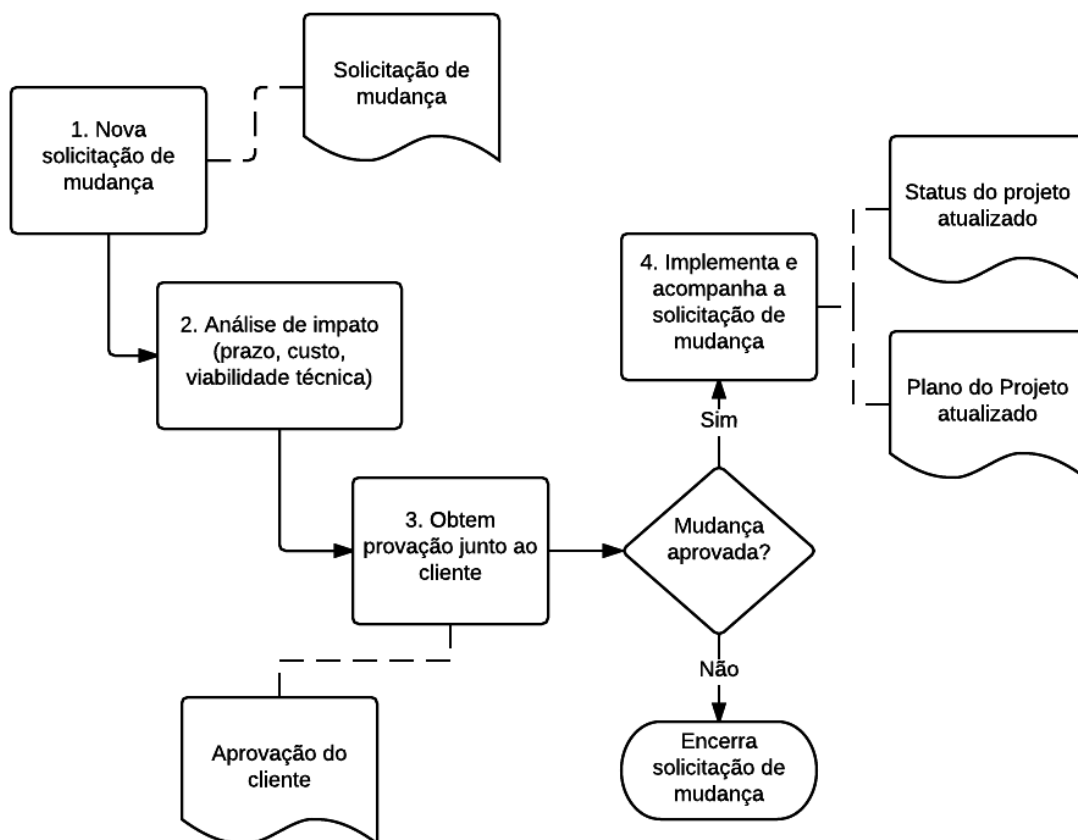
Objetivo: PM.O2, PM.O3, PM.O4, PM.O5, PM.O7

4.4.8 Registrar e Acompanhar Mudanças

O objetivo desta tarefa é analisar e avaliar a solicitação de mudança quanto ao seu impacto no custo, prazo e viabilidade técnica. A solicitação de mudança pode ser iniciada externamente, pelo cliente, ou internamente, pela equipe do projeto. Caso a solicitação de mudança não afete os acordos com o cliente, o Plano do Projeto deverá ser atualizado. Caso contrário, o cliente deverá ser comunicado e a mudança deverá ser negociada. Sua implementação e consequente alteração no Plano do Projeto deverão iniciar mediante aprovação do cliente.

Com base nas recomendações do grupo IT Project Management (2007), esta tarefa segue o seguinte fluxo de atividades:

Figura 17 - Fluxo de atividades da solicitação de mudança



Fonte: Autor

Produtos de Entrada: Solicitação de mudança

Produtos de Saída: Registro de aceite, Plano do Projeto (atualizado)

Papel: Gerente de Projetos (PM), Cliente (CUS), Equipe do projeto (WT)

Atividade: PM.3

Objetivo: PMO.2 e PM.03

4.4.9 Validação

Como descrito anteriormente, a Validação visa garantir que o software desenvolvido atende ao uso pretendido do software, definidos junto ao cliente.

A tarefa de Validação constrói e mantém uma matriz de rastreabilidade que relaciona as funcionalidades implementadas no software com os requisitos definidos formalmente na Especificação dos Requisitos de Software. Uma funcionalidade de software pode ser uma tela, algoritmo, módulo de software ou qualquer funcionalidade que atenda um ou mais requisitos do produto. A figura 18 exibe um exemplo de matriz de rastreabilidade.

Figura 18 - Matriz de rastreabilidade dos requisitos

	Funcionalidade X	Funcionalidade Y	Funcionalidade Z
Requisito 01	X		X
Requisito 02		X	
Requisito 03		X	
Requisito 04			X

Fonte: Autor

O responsável por conduzir esta atividade é o Analista (AN). Como responsável pela Especificação de Requisitos de software, o Analista é o representante da Equipe do Projeto (WT) mais indicado para representar o cliente, e garantir a aderência do produto em relação aos requisitos. Além disto, o Analista deve garantir que o comportamento do software esteja de acordo com o pretendido (por exemplo, um software que gera folha de pagamentos deverá fornecer uma relação de funcionários e seus respectivos salários conforme regras definidas na Especificação de Requisitos).

Produtos de Entrada: Declaração de Trabalho, Especificação dos Requisitos de Software, Software

Produtos de Saída: Registro de validação

Papel: Analista (AN)

Atividade: SI.5

Objetivo: SI.O7

4.4.10 Encerrar o Projeto

Esta atividade formaliza o termino do projeto de acordo com as instruções de configuração definidas no plano do projeto.

O Encerramento do Projeto define o processo de aceite e entrega do software ao cliente. A seguir estão relacionadas as principais informações para entrega do projeto:

- Gerar baseline de acordo com os módulos que compõe o produto;
- Configuração do Software: presente no Plano do Projeto, define a estratégia de entrega (por exemplo: versão, ambiente, nomenclatura, requisitos de infraestrutura);
- Aceite do cliente: formaliza a entrega do software, comprova que o produto de software atende ao uso pretendido definido pelos Stakeholders;
- Registro de lições aprendidas no projeto;
- Realizar um backup do repositório do projeto. Remover o acesso ao repositório da Equipe do Projeto (WT);

Produtos de Entrada: Software, Plano do Projeto, Aceite do cliente

Produtos de Saída: Software (entregue)

Papel: Gerente de projetos (PM), Líder Técnico (TL)

Atividade: PM.4

Objetivo: PMO.2

4.5 Considerações do Capítulo

Neste capítulo, o estudo de caso sobre implantação de boas práticas em pequenas empresas de software foi descrito. Um novo processo de desenvolvimento aderente ao processo Gerenciamento de Projetos da norma ISO/IEC 29110-4 foi descrito e aplicado em uma VSE.

Com base nisto, o próximo capítulo descreve os resultados do estudo de caso, bem como propostas para trabalhos futuros.

5. CONSIDERAÇÕES FINAIS

Este capítulo apresenta o resultado do estudo de caso, onde o processo de gerenciamento de projetos descrito no capítulo anterior foi implantado em uma pequena empresa e usado durante um projeto real.

Além disso, este capítulo apresenta algumas sugestões para trabalhos futuros, desta forma encerrando este trabalho.

5.1 Contribuições do Trabalho

Após a escolha da norma ISO/IEC 29110-4, um processo de desenvolvimento aderente à norma foi criado e documentado. Somente o processo de Gerenciamento de Projetos foi implementado.

Este processo, aderente às boas práticas de engenharia de software, foi implementado em uma pequena empresa. O processo foi adotado em um projeto de baixa complexidade e pequeno em duração, cuja estimativa seria de 30 dias. Os seguintes papéis foram treinados e participaram do projeto:

- Gerente de Projetos (GP);
- Designer (DES);
- Líder Técnico (TL);
- Analista (AN);
- Programador (PR);

Facilitaram a implantação do novo processo:

- A equipe já executava algumas atividades que puderam ser aproveitadas no novo processo. Destas, podemos relacionar o repositório central, o registro de reunião e o registro de status do projeto;
- A equipe não apresentou queixas ou resistência à mudança. Todos se mostraram entusiasmados pelo novo processo, por acreditar em sua capacidade de diminuir o retrabalho e aumentar a qualidade das entregas;

Dificultaram a implantação do processo:

- Obter a colaboração do cliente em relação ao aceite formal da Declaração de Trabalho, Plano do Projeto e Registro de aceite para entrega do software. O processo foi criticado negativamente e qualificado como burocrático;

Apesar da crítica negativa, o processo foi avaliado positivamente pela equipe do projeto. De acordo com a visão da equipe, a formalização do aceite, criticada pelo cliente, foi justamente o que contribuiu para a entrega do produto com qualidade e com diminuição do retrabalho.

Durante o curso do projeto, foram realizadas duas solicitações de mudança, ambas por parte do cliente. A primeira delas surgiu durante a reunião de alinhamento do status do projeto junto ao cliente e tinha como objetivo adicionar uma funcionalidade ao produto. A equipe analisou o impacto e o risco da mudança com relação ao custo e ao prazo do projeto. O impacto foi apresentado ao cliente em reunião subsequente e, devido ao aumento do prazo, foi abandonado.

A segunda solicitação de mudança, referente a uma mudança na regra de negócio de um requisito já existente, também surgiu em uma reunião de alinhamento do status do projeto. Neste caso, o impacto em prazo e risco foram analisados, considerados irrelevantes, e foram incorporados ao projeto. O Plano do projeto foi atualizado e o cliente foi notificado da mudança.

Os seguintes pontos foram levantados durante a coleta de informações a respeito do estudo de caso:

- O Gerente de Projetos realizou um total de 8 registros de Status do Projeto, dois registros por semana;
- As reuniões de alinhamento com o cliente foram realizadas semanalmente, portanto ao todo quatro reuniões foram realizadas. Estas reuniões resultaram em solicitações de mudança (descritas anteriormente) e uma solicitação de correção.
- Durante a construção do software, diversas dúvidas foram levantadas pelo programador. Estas dúvidas foram devidamente levadas ao cliente durante as reuniões de alinhamento e registradas em ata de reunião;

- A etapa da Validação forneceu a matriz de rastreabilidade de requisitos. Nenhum requisito deixou de ser implementado e a matriz de rastreabilidade garantiu que todos os requisitos de software estivessem cobertos pelas funcionalidades do software;
- O software foi entregue dentro do prazo e do custo estimados, a reunião de apresentação do software foi realizada e o aceite do software foi concedido. Esta reunião não produziu solicitações de ajustes ou correções no software;
- Ao término do projeto e posterior entrega do software, iniciou-se a etapa de manutenção, ficando a cargo também da LogSoft. Os produtos de trabalho gerados durante este estudo de caso foram julgados de grande importância para a equipe de manutenção. O processo anterior não garantia documentação e a equipe de manutenção buscava informações diretamente com o(s) programador(es) presente(s) durante o projeto. Os produtos de trabalho gerados a partir deste estudo de caso garantiram que as informações do produto e do projeto estivessem disponíveis para a equipe de manutenção;

Após o termino do projeto, foi feita uma pesquisa de satisfação junto ao cliente. A tabela 19 apresenta o roteiro de perguntas e as respectivas respostas.

Tabela 19 - Resultados da pesquisa de satisfação

Perguntas	Cliente
O software atendeu às expectativas? (Totalmente / Razoavelmente / Negativamente)	Sim, atendeu totalmente
Qual sua percepção em relação a qualidade do produto? (Ótimo / Bom / Regular / Ruim)	Bom
Qual sua percepção em relação a qualidade do processo de desenvolvimento? (Ótimo / Bom / Regular / Ruim)	Bom
Foi entregue dentro do prazo e dos custos previamente acordados?	Sim
Qual sua opinião a respeito da aprovação formal da Declaração de Trabalho, Plano do Projeto e o Registro de aceite do projeto?	Acredito que a assinatura adicionou maior burocracia, porém também trouxe maior clareza a respeito do que estava sendo desenvolvido e do que foi entregue.
Voltaria a adquirir o produto e/ou serviços da LogSoft?	Sim

Fonte: Autor

De acordo com o Gerente de TI da LogSoft, o novo processo introduziu grandes melhorias para a equipe de desenvolvimento. A empresa vai expandir o processo aqui descrito, para os outros projetos. O processo SI (Desenvolvimento de Software) será implementado, desta forma completando a adoção das boas práticas descritas pela norma ISO/IEC 29110-4. O objetivo é obter a certificação e ser reconhecida internacionalmente como uma empresa produtora de software com qualidade.

Desta forma, encerra-se este trabalho, apresentando os resultados do estudo de caso. Espera-se que as boas práticas apresentadas neste trabalho possam auxiliar as pequenas empresas produtoras de software que estejam buscando maior qualidade em seu processo de software e maior competitividade.

5.2 Trabalhos Futuros

Este trabalho teve como escopo a implementação do processo PM, portanto uma sugestão para um trabalho futuro seria a implementação do processo SI, desta forma implementando completamente o padrão de processo proposta pela norma ISO/IEC 29110-4.

Outra sugestão para um trabalho futuro é um estudo da norma ISO/IEC 29110 sob o ponto de vista das equipes focadas em manutenção de software. Muitas VSEs são responsáveis por manter o software após o término do projeto, realizando alterações e mudanças para se adequar às novas necessidades do cliente.

REFERÊNCIAS

ABES: MERCADO BRASILEIRO DE SOFTWARE PANORAMA E TENDÊNCIAS 2014 - ISBN 978-8586700-03-3.

CAPABILITY Maturity Model Integration (CMMI). Version 1.3. Carnegie Mellon: Software Engineering Institute, 2010.

CHRISSIS, M. B., KONRAD, M, E SHRUM, S. CMMI: Guidelines for Process Integration and Product Improvement, 2003.

CMMMI Institute, Global adoption of CMMI, disponível em <http://cmmiinstitute.com/newsroom/global-adoption-cmmi-performance-improvement-framework-continues-rise>, acessado em 06/01/2015.

IEEE, The management of software engineering, Part I: Principles of software engineering, disponível em <http://ieeexplore.ieee.org/>, acessado em 23/07/2015.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO/IEC 29110-4: Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 4-1: Profile specifications: Generic profile group. Geneva, 2011b.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO/IEC TR 29110-5-1-2: Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile. Geneva, 2011c.

IT Project Management, The Most Common Reasons for Change Requests disponível em <http://it-project-guide.blogspot.com.br/2007/09/most-common-reasons-for-change-requests.html>, acessado em 11/02/2015.

Laporte, C. Y: Public Site of the ISO Working Group Mandated to Develop ISO/IEC 29110 Standards and Guides for Very Small Entities involved in the Development or Maintenance of Systems and/or Software, Prof Claude Y, disponível em <http://profs.etsmtl.ca/claporte/english/vse/>, acessado em 03/12/2014.

Pessoa, M., Gonzalez A. Experiencia Brasileña en el uso de la ISO/IEC 29110 2015.

PINHEIRO, Diogo Graça: Iniciando o CMMI em uma pequena empresa de engenharia de software, 2010.

POSCA, R. B., HIRAMA, K.: UMA ANÁLISE DE COBERTURA DA NORMA ISO/IEC 29110 EM RELAÇÃO AOS MODELOS MR-MPS-SW E CMMI-DEV, 2015

PRESSMAN, Roger S. Software Engineering A Practitioner's Approach 7th Edition. McGraw-Hill, 2009.

SEBRAE: Pequenas empresas no PIB, disponível em <http://www.sebrae.com.br/sites/PortalSebrae/ufs/mt/noticias/Micro-e-pequenas-empresas-geram-27%25-do-PIB-do-Brasil>, acessado em 31/05/2015.

SEI, Software Engineering Institute. Disponível em <http://www.sei.cmu.edu/>, acessado em 07/04/2015.

SOFTEX. Guia Geral MPS de Software. 2012. Disponível em: http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012-c-ISBN-1.pdf, acessado em 21/03/2015.

SOMMERVILLE, IAN. Engenharia de Software. 8a. edição, Addison-Wesley, 2007.

GLOSSÁRIO

Stakeholders – termo usado para se referir às partes interessadas no sucesso do projeto, por exemplo gestores do negócio, desenvolvedores de software, usuários finais.

Startups – pequenas empresas iniciantes, tipicamente relacionadas a área de tecnologia. Possuem um forte espírito empreendedor, e uma constante busca por um modelo de negócio inovador.

Baseline – especificação ou produto que tenha sido formalmente revisado e acordado, que posteriormente serve de base para o desenvolvimento, e que só pode ser alterado por meio de procedimentos formais de controle de mudanças.