

UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS

DIEGO AUGUSTO ERTEL

**IMPLEMENTAÇÃO DE CONCEITOS E PRÁTICAS DE FÁBRICA DE SOFTWARE
NO PROCESSO DE DESENVOLVIMENTO**

BENTO GONÇALVES

2020

DIEGO AUGUSTO ERTEL

**IMPLEMENTAÇÃO DE CONCEITOS E PRÁTICAS DE FÁBRICA DE SOFTWARE
NO PROCESSO DE DESENVOLVIMENTO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias, Campus Universitário da Região dos Vinhedos, da Universidade de Caxias do Sul.

Orientadora: Profa. Ma. Iraci Cristina da Silveira De Carli

BENTO GONÇALVES

2020

DIEGO AUGUSTO ERTEL

**IMPLEMENTAÇÃO DE CONCEITOS E PRÁTICAS DE FÁBRICA DE SOFTWARE
NO PROCESSO DE DESENVOLVIMENTO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias, Campus Universitário da Região dos Vinhedos, da Universidade de Caxias do Sul.

Aprovado em 02 / 12 / 2020

Banca examinadora:

Profa. Ma. Iraci Cristina da Silveira De Carli
Universidade de Caxias do Sul – UCS

Prof. Me. Giovanni Ely Rocco
Universidade de Caxias do Sul - UCS

Prof. Dra. Maria de Fátima Webber do Prado Lima
Universidade de Caxias do Sul – UCS

AGRADECIMENTOS

Primeiramente agradeço à Deus, pela vida e pelas oportunidades que tenho recebido. Por me iluminar e me dar forças para lutar pelos meus objetivos pessoais e profissionais.

Agradeço a minha família, mãe e irmãos, que estão presentes em todos os momentos e acompanharam minha trajetória até aqui.

Agradeço a minha professora orientadora, Iraci Cristina da Silveira De Carli, pela sua disponibilidade, atenção e dedicação. Agradeço pela ajuda e por acreditar em mim.

Por fim, agradeço a todos os colegas, amigos e professores que contribuíram de alguma forma para a realização deste trabalho.

RESUMO

Como não existe um consenso sobre um processo ideal de desenvolvimento de software, diversos pontos do processo atual de desenvolvimento de uma organização podem ser melhorados. Alterar o processo de desenvolvimento é uma tarefa complexa, e não significa simplesmente adotar métodos, ferramentas ou modelos de processos padronizados, pois questões técnicas, fatores internos, culturais e humanos também devem ser analisados e considerados. Um aspecto muito importante neste sentido é a evolução frequente dos sistemas desenvolvidos, uma vez que constituem os principais ativos de negócio para a organização. O conceito Fábrica de Software consiste em simplificar, padronizar e aperfeiçoar processos, eliminar o que não agrega valor, reutilizar componentes, diminuir o tempo de execução e custos, além de aumentar a eficiência e a qualidade do que é desenvolvido. Desta forma, o presente trabalho tem por objetivo estudar, analisar e implementar conceitos e boas práticas de fábrica de software no atual processo de desenvolvimento da organização estudada.

Palavras-chave: Fábrica de Software.

ABSTRACT

Since there is no consensus on an ideal software development process, several points in an organization current development process can be improved. Changing the development process is a complex task, and does not mean simply adopting standardized methods, tools or process models, technical issues, internal, cultural and human factors must also be analyzed and considered. A very important aspect in this regard is the frequent evolution of the systems developed, since they are the main business assets for the organization. The Software Factory concept consists of simplifying, standardizing and improving processes, eliminating what does not add value, reusing components, reducing execution time and costs, in addition to increasing the efficiency and quality of what is developed. Thus, the present work aims to study, analyze and implement software factory concepts and good practices in the current development process of the studied organization.

Keywords: Software Factory.

LISTA DE FIGURAS

Figura 1 - Processo de mudança do processo	31
Figura 2 - Processo da XP	35
Figura 3 - Ciclo de vida Adaptive.....	39
Figura 4 - Loop de aprendizado	39
Figura 5 - Modelo do atual processo de desenvolvimento	62
Figura 6 - Fases e artefatos do processo de desenvolvimento proposto	66
Figura 7 - Processo proposto para a fase de planejamento	67
Figura 8 - Exemplo do artefato Histórias	69
Figura 9 - Processo executado na fase de desenvolvimento	70
Figura 10 - Processo executado na fase de entrega	72

LISTA DE QUADROS

Quadro 1 - Evolução das fábricas de software por fases	21
Quadro 2 - Comparativo entre um processo maduro e um processo imaturo	23
Quadro 3 - Evolução do desenvolvimento de software	28
Quadro 4 - Áreas-chave por nível do CMM	43
Quadro 5 - Áreas-chave por estágio do CMMI	45
Quadro 6 - Categorização das áreas dos processos	46
Quadro 7 - Resumo das questões e métricas por objetivo.....	52
Quadro 8 - Respostas por objetivos e questões.....	56
Quadro 9 - Respostas obtidas na entrevista	59
Quadro 10 - Listas de funcionalidades do projeto Benefícios oferecidos pelo Espaço Bem-Estar	79
Quadro 11 - Lista de funcionalidades do projeto Registrar os treinamentos de funcionários temporários	80
Quadro 12 - Lista de funcionalidades do projeto Cadastro dos candidatos via formulário	80
Quadro 13 - Lista de atividades do projeto Benefícios oferecidos pelo Espaço Bem-Estar.....	81
Quadro 14 - Lista de atividades do projeto Registrar os treinamentos de funcionários temporários	81
Quadro 15 - Lista de atividades do projeto Cadastro dos candidatos via formulário ..	81
Quadro 16 - Horas de trabalho do projeto Benefícios oferecidos pelo Espaço Bem-Estar.....	82
Quadro 17 - Horas de trabalho do projeto Registrar os treinamentos de funcionários temporários	83
Quadro 18 - Horas de trabalho do projeto Cadastro dos candidatos via formulário ..	83
Quadro 19 - Métricas aplicadas durante o estudo de caso	85
Quadro 20 - Cálculos e resultados por métrica	88

LISTA DE SIGLAS

AEDB	Associação Educacional Dom Bosco
ASAP	Accelerated SAP
ASD	Adaptive Software Development
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CRC	Class Responsibility Collaboration Card
DH	Desenvolvimento Humano
ER	Entidade-Relacionamento
FSA	Fábrica de Software Acadêmica
GQM	Goal-Question-Metric
GT	Grupos de Trabalho
IES	Instituições de Ensino Superior
LD	Lean Development
MDSI	Metodologia de Desenvolvimento de Sistemas e Integração
PCP	Planejamento e Controle da Produção
PMI	Project Management Institute
RH	Recursos Humanos
SPL	Software Product Lines
XP	Extreme Programming

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	PROBLEMA DE PESQUISA.....	12
1.1.1	Questão de pesquisa	12
1.2	JUSTIFICATIVA.....	13
1.3	OBJETIVOS	13
1.3.1	Objetivo geral	13
1.3.2	Objetivos específicos	14
1.4	METODOLOGIA.....	14
1.5	ESTRUTURA DO TEXTO.....	15
2	FÁBRICA DE SOFTWARE.....	17
2.1	FÁBRICAS TRADICIONAIS X FÁBRICAS DE SOFTWARE	18
2.2	ORGANIZAÇÃO DA FÁBRICA DE SOFTWARE.....	19
2.3	EVOLUÇÃO DA FÁBRICA DE SOFTWARE POR FASES	20
2.4	ATRIBUTOS DA FÁBRICA DE SOFTWARE	21
2.5	MATURIDADE DA FÁBRICA DE SOFTWARE.....	23
2.6	TRABALHOS CORRELATOS.....	24
2.6.1	Definição e estabelecimento de processos de fábrica de software em uma organização de TI do setor público	24
2.6.2	Um modelo de Fábrica de Software em instituições de ensino superior.....	25
2.7	CONSIDERAÇÕES FINAIS.....	26
3	PROCESSOS E METODOLOGIAS UTILIZADOS POR UMA FÁBRICA DE SOFTWARE	27
3.1	HISTÓRICO	28
3.2	APRIMORAMENTO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE ..	29
3.2.1	Medir	29
3.2.2	Analisar.....	30
3.2.3	Mudar.....	31
3.3	PROCESSOS TRADICIONAIS.....	32
3.3.1	Método Cascata (Waterfall)	32
3.4	PROCESSOS ÁGEIS.....	33
3.4.1	Software Product Lines	33

3.4.2	Accelerated SAP	34
3.4.3	Extreme Programming	35
3.4.4	Adaptive Software Development	38
3.4.5	Lean Development	41
3.5	MODELOS E PROCESSOS DE MATURIDADE	42
3.5.1	Capability Maturity Model	42
3.5.2	Capability Maturity Model Integration	43
3.6	CONSIDERAÇÕES FINAIS	46
4	MEDIÇÃO E ANÁLISE DO PROCESSO ATUAL	48
4.1	MEDIÇÃO DO PROCESSO ATUAL	49
4.1.1	Definir e documentar um processo padronizado de desenvolvimento	50
4.1.2	Deixar mais claro o processo de análise de requisitos	50
4.1.3	Aumentar a eficiência da equipe de desenvolvimento	51
4.1.4	Aumentar a satisfação do cliente	51
4.2	ANÁLISE DO PROCESSO ATUAL	53
4.2.1	Questionário	53
4.2.2	Entrevistas	58
4.2.3	Observação	59
4.2.4	Modelo do processo atual	61
4.3	CONSIDERAÇÕES FINAIS	63
5	APRIMORAMENTO DO PROCESSO ATUAL	64
5.1	IDENTIFICAR OS APRIMORAMENTOS DO PROCESSO	64
5.2	O NOVO MODELO DO PROCESSO DE DESENVOLVIMENTO	66
5.2.1	A fase de Planejamento	67
5.2.2	A fase de Desenvolvimento	70
5.2.3	A fase de Entrega	72
5.3	CONSIDERAÇÕES FINAIS	73
6	ESTUDO DE CASO	75
6.1	DESCRIÇÃO DO ESTUDO DE CASO	75
6.1.1	Projeto Benefícios oferecidos pelo Espaço Bem-Estar	76
6.1.2	Projeto Registrar os treinamentos de funcionários temporários	76
6.1.3	Projeto Cadastro dos candidatos via formulário	77

6.2	APLICAÇÃO DA METODOLOGIA PROPOSTA.....	78
6.2.1	Fase de planejamento.....	79
6.2.2	Fase de desenvolvimento.....	82
6.2.3	Fase de entrega.....	84
6.2.4	Medição do processo	84
6.3	CONSIDERAÇÕES FINAIS.....	88
7	CONCLUSÃO.....	91
	REFERÊNCIAS	95
	APÊNDICE A - Questionário aplicado na análise do processo atual	97
	APÊNDICE B - Entrevista aplicadas na análise do processo atual	99
	APÊNDICE C - Processo padronizado aplicado no estudo de caso	102
	APÊNDICE D - Entrevista aplicada após a aplicação do estudo de caso	104

1 INTRODUÇÃO

A produção de software surgiu em 1960, de forma artesanal e ineficiente em termos de produtividade, redução de custos e tempo de execução.

Segundo Tenório e Valle (2013), o conhecimento sobre determinado sistema ou aplicação ficava retido com o desenvolvedor, fazendo com que a manutenção e a implementação de novas funcionalidades dependessem do mesmo para serem efetuadas.

A partir da revolução industrial, baseada no taylorismo, no fordismo e no pós-fordismo, o processo de desenvolvimento de software passou a ser pensado como uma linha de produção, assim como nas fábricas tradicionais, com a organização da produção em esteiras, a divisão do processo em etapas, a cronometragem do tempo de execução e a redução de custos. Segundo os autores, mesmo que as fábricas de software tenham adotado conceitos utilizados na indústria, o processo de desenvolvimento é totalmente distinto. Enquanto que nas fábricas tradicionais os produtos são padronizados e produzidos em larga escala, nas fábricas de software cada produto é único e somente alguns componentes são reutilizados. Na indústria, o colaborador necessita conhecer sua etapa e saber desempenhar sua função, enquanto que nas fábricas de software o conhecimento deve ser passado para a etapa seguinte e, quanto maior é o conhecimento do colaborador sobre o processo como um todo, maior será sua eficiência ao executar as tarefas.

O conceito Fábrica de Software tem o objetivo de estabelecer um modelo que minimize custos, aumente a produtividade e melhore a qualidade no desenvolvimento de sistemas ou produtos de software. O mesmo pode ser empregado em empresas independentes ou em setores de grandes empresas. Sua concepção é baseada em padrões de processos definidos, melhoria contínua, alta eficiência, garantia e controle da qualidade, reaproveitamento de componentes, perfis funcionais adequados às funções e boa comunicação. Para tal, utiliza ferramentas de apoio e boas práticas, baseadas em planejamento adequado, padronização de processos, metodologias ágeis, gerenciamento da qualidade, gerenciamento de recursos humanos, gerenciamento de riscos e comunicação.

1.1 PROBLEMA DE PESQUISA

Em empresas de desenvolvimento de software, problemas que ocorrem frequentemente são a dificuldade de entendimento das necessidades do cliente, falhas nos processos de análise de requisitos e de comunicação, falta de processos definidos para a garantia e o controle da qualidade, falhas no gerenciamento de riscos, alto tempo gasto em manutenção, a falta de eficiência, além de altos índices de erros e retrabalho, gerando o não cumprimento dos prazos estipulados inicialmente.

Atualmente, a equipe de desenvolvimento do Departamento de Informática da empresa Tramontina Central de Administração, desenvolve cerca de 90% dos sistemas que são utilizados pela organização. Cada colaborador é responsável por alguns dos sistemas e atua em todas as etapas do processo de desenvolvimento, desde o levantamento de requisitos até os testes e a implantação. Um dos principais problemas do processo atual é que não há uma metodologia com modelagem, definição de escopo, controle de tarefas e tempo definido para execução. Há somente um programa gerenciador de projetos, que possui o ano e mês de planejamento para a entrega de cada projeto. A definição deste ano e mês é baseada em uma análise prévia e na experiência do responsável pelo sistema. Diversas vezes, o prazo para a entrega não é cumprido. Além disso, a eficiência no desenvolvimento é medida pela quantidade de projetos encerrados em cada sistema, o que é muito relativo, pois há projetos de baixa, média e alta complexidade.

Sendo assim, o problema de pesquisa deste trabalho foi realizar um estudo sobre a implementação de conceitos e boas práticas de fábrica de software, melhorando assim o processo de desenvolvimento da empresa.

1.1.1 Questão de pesquisa

A implantação de conceitos e boas práticas de fábricas de software pode contribuir na melhoria do processo de desenvolvimento e, conseqüentemente, dos softwares desenvolvidos?

1.2 JUSTIFICATIVA

O processo de desenvolvimento do Departamento de Informática da empresa Tramontina Central de Administração atualmente acarreta em problemas para a empresa, pois várias vezes o tempo de desenvolvimento fica acima do planejado e a eficiência é medida pela quantidade de projetos encerrados, medindo a produtividade pela quantidade e desconsiderando a complexidade de cada projeto.

Portanto, buscou-se através deste trabalho realizar um estudo sobre a implementação de conceitos e boas práticas de fábrica de software no atual processo de desenvolvimento da empresa, fazendo com que haja um processo padronizado, que efetue o controle de todas as etapas, desde o levantamento de requisitos junto ao cliente, até a entrega do produto, com prazos definidos para a execução de cada tarefa. Além disso, o objetivo foi que esse processo padronizado, pudesse ser implementado em outras empresas. As principais hipóteses levantadas foram o reaproveitamento de componentes, a melhoria do gerenciamento da qualidade, a melhoria da comunicação entre os processos e a implantação de uma nova forma de verificar a eficiência no processo de desenvolvimento.

1.3 OBJETIVOS

Os objetivos geral e específicos deste trabalho são apresentados a seguir.

1.3.1 Objetivo geral

Este trabalho teve como objetivo geral realizar um estudo de caso sobre a implementação de boas práticas de fábrica de software no Departamento de Informática da empresa Tramontina Central de Administração. A aplicação e validação do objetivo foi realizada em um piloto, com a participação de um número reduzido de colaboradores.

1.3.2 Objetivos específicos

Para atender ao objetivo geral, este trabalho propôs os seguintes objetivos específicos:

- Elaboração de um referencial teórico, baseado em pesquisas bibliográficas;
- Estudo de casos correlatos de estudo e implantação de conceitos e boas práticas de fábrica de software;
- Análise do atual processo de desenvolvimento da organização;
- Definição de critérios de avaliação das melhorias a serem implementadas;
- Proposta de aspectos e melhorias identificados;
- Aplicação das melhorias em um grupo piloto, avaliando os resultados obtidos com a implantação dos aspectos e melhorias identificados.

1.4 METODOLOGIA

A metodologia do trabalho consistiu inicialmente em elaborar um referencial teórico, baseado em pesquisas e bibliografias, sobre os conceitos e boas práticas que norteiam uma fábrica de software, a fim de identificar aspectos positivos e aspectos críticos.

Após a elaboração do referencial teórico, foram estudados casos correlatos de implementação dos conceitos e boas práticas, a fim de identificar possíveis semelhanças de aplicação.

Partindo deste levantamento, foi realizado o estudo e o detalhamento do atual processo de desenvolvimento do Departamento de Informática da empresa Tramontina Central de Administração, utilizando técnicas de análise sugeridas por Sommerville (2007), como questionário, entrevistas e observação, e o método de análise, medição e aprimoramento do processo GQM – (Objetivo-Questão-Métrica), que também é sugerido pelo autor.

Tendo o atual processo desenhado e detalhado, foram definidos os critérios de avaliação da abordagem proposta e foi criado um esboço dos aspectos de mudança na abordagem de desenvolvimento atual. Através dos aspectos de mudança, foi realizado um estudo de caso com experimento em um piloto, utilizando uma pequena quantidade de desenvolvedores, para verificar se as alterações propostas resultaram em melhoras no processo de desenvolvimento. A validação, bem como a eficácia das melhorias propostas, foram efetuadas através de critérios qualitativos.

1.5 ESTRUTURA DO TEXTO

O texto foi estruturado em 7 capítulos, onde no Capítulo 1 é apresentada a introdução deste trabalho, composta pela contextualização do assunto, pelo problema de pesquisa, pela justificativa, pelos objetivos geral e específicos e pela metodologia.

No Capítulo 2 são apresentados o conceito de Fábrica de Software, o comparativo entre as fábricas tradicionais e de software, além de características da fábrica, evolução por fases da mesma, os atributos, o conceito de maturidade, trabalhos correlatos e as considerações finais.

O Capítulo 3 traz um breve histórico da evolução dos processos e metodologias de desenvolvimento utilizadas pelas fábricas de software ao longo do tempo. Além disso, realiza um estudo sobre conceitos, processos, métodos e técnicas que podem colaborar na implementação de uma fábrica de software, tais como: aprimoramento de processo, processos tradicionais, processos ágeis e níveis de maturidade.

No Capítulo 4, é realizada a análise e medição do atual processo de desenvolvimento da organização utilizada como estudo de caso deste trabalho. Após o levantamento do atual processo, são definidos os objetivos, questões e métricas que serão consideradas para a proposta de implementação.

O Capítulo 5 traz a proposta de implementação de melhorias no atual processo de desenvolvimento da organização utilizada como estudo de caso deste trabalho, baseada nos objetivos, questões e métricas definidos anteriormente.

No Capítulo 6 é apresentada a aplicação do estudo de caso proposto neste trabalho, bem como os resultados obtidos através de sua aplicação.

Por fim, o Capítulo 7 traz as considerações finais sobre o estudo realizado.

2 FÁBRICA DE SOFTWARE

A expressão “Fábrica de Software” é utilizada desde os anos 60, porém o conceito surgiu apenas em meados dos anos 80. O mesmo é oriundo de modelos implantados na revolução industrial, como o taylorismo, o fordismo e o pós-fordismo, pois era necessária a produção em larga escala, com alta qualidade e baixo custo.

Fernandes e Teixeira (2004, p. 117) definem o conceito da seguinte forma:

Processo estruturado, controlado e melhorado de forma contínua, considerando abordagem de engenharia industrial, orientado para o atendimento a múltiplas demandas de natureza e escopo distintas, visando à geração de produtos de software, conforme os requerimentos documentados dos usuários e/ ou clientes, da forma mais produtiva e econômica possível.

Para Greenfield¹ (2003 apud TENÓRIO; VALLE, 2013), o conceito pressupõe o desenvolvimento de sistemas baseado em componentes, direcionado a modelos e a linhas de produto de software, o que caracteriza uma iniciativa de fábrica, barateando a montagem de aplicações por conta da reutilização sistemática, que possibilita a formação de cadeias de produção.

Na prática, o conceito baseia-se em estabelecer um método, um processo gerencial e/ou produtivo efetivo, com escopo, agregação de valor ao negócio, previsibilidade no atendimento dos serviços, baixo custo, segurança e alta qualidade. Os princípios de aplicação do conceito podem ser descritos da seguinte forma:

- Eliminação do que não agrega valor;
- Simplificação, automatização e aperfeiçoamento dos processos;
- Aplicação consistente de padrões;
- Reutilização dos componentes;
- Diminuição do tempo de execução e dos custos;
- Aumento da qualidade.

Para Fernandes e Teixeira (2004, p. 116), o objetivo da Fábrica de Software deve ser “a geração de produtos requeridos pelos usuários e/ ou clientes, com o mínimo de defeitos possível e a um preço (ou custo) competitivo e compatível que

¹ GREENFIELD J.; SHORT, K. Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools. **Indianápolis: Wiley**, 2003.

forneça a margem necessária para os investimentos em manutenção e melhoria da fábrica”.

Este capítulo apresenta um comparativo entre fábricas tradicionais e de software. Em seguida, são apresentados aspectos referentes à organização da fábrica de software, fases, atributos característicos e níveis de maturidade. O capítulo é encerrado com os trabalhos correlatos e as considerações finais.

2.1 FÁBRICAS TRADICIONAIS X FÁBRICAS DE SOFTWARE

Conforme Tenório e Valle (2013), apesar do processo produtivo das fábricas de software seguir conceitos de operações utilizadas na indústria, o desenvolvimento de um sistema ou produto de software é completamente distinto em relação aos processos de fabricação tradicionais. Mesmo que exista uma esteira por onde passam os componentes, a mesma é invisível e o produto software é de natureza abstrata durante quase todo o ciclo de desenvolvimento. Cada software é um produto único e distinto, enquanto que nas operações das fábricas convencionais as técnicas são repetitivas e o produto é produzido de forma padronizada e em larga escala.

Nas fábricas tradicionais, basta ao funcionário conhecer as técnicas, as ferramentas a serem utilizadas e a peça que o mesmo deve fabricar, não sendo fundamental conhecer o produto e o contexto de sua utilização. Contudo em fábricas de software, todo o conhecimento gerado em cada etapa deve ser registrado e transmitido para as etapas seguintes. Portanto, ao executar a próxima tarefa, o colaborador deve recorrer a uma “base de conhecimentos” dinâmica, compartilhada e atualizada. Trafegam pelas esteiras tanto os artefatos quanto o conhecimento. Quanto maior for o conhecimento da equipe de desenvolvimento, maior será a eficiência na execução das tarefas.

Ainda segundo os autores, em fábricas de software as tarefas não são totalmente repetitivas, diferentemente das fábricas convencionais, pois existem etapas que necessitam de criatividade para serem executadas. Muitas delas, inclusive, necessitam de discussão para decidir a melhor solução a ser adotada. As esteiras das linhas de montagem fordistas, por exemplo, não previam retornos das

peças às etapas anteriores, para casos de processos defeituosos ou inacabados. Quando ocorria um problema em uma das etapas, toda a produção era prejudicada. Considerando que um sistema ou software sofre constantes mudanças de regras e escopo, as “linhas de montagem” de uma fábrica de software devem ser flexíveis o bastante para que o produto retorne à etapa anterior, sem que ocorra o bloqueio da produção.

Nas fábricas tradicionais, mesmo que existam formas automatizadas de medição de desempenho, eficiência e qualidade, o controle da produção e dos colaboradores é efetuado por um supervisor de produção. Nas fábricas de software, todo o controle é realizado por sistemas de apoio informatizados e o supervisor desempenha funções estratégicas.

2.2 ORGANIZAÇÃO DA FÁBRICA DE SOFTWARE

Para Araújo² (2007, p. 150 apud TENÓRIO; VALLE, 2013), “estruturação é uma forma sistematizada de agrupar atividades em frações organizacionais definidas seguindo um dado critério, visando a melhor adequação da estrutura organizacional e sua dinâmica de ação”.

Em uma fábrica de software, as equipes trabalham em grupos e em espaços delimitados por ilhas ou divisórias, onde cada membro utiliza seu computador. O processo de desenvolvimento do produto é que determina a variação dos integrantes destes grupos. A esteira de produção está presente, porém é difícil identificar visualmente quais são as equipes, quais são as etapas, em qual etapa cada equipe atua, ou ainda o papel de cada colaborador no processo de desenvolvimento. Fatores importantes na organização da fábrica é que a tecnologia empregada deve permitir que os funcionários permaneçam trabalhando nos seus locais de trabalho, independentemente da equipe ou software na qual os mesmos estão trabalhando, e que os colaboradores possam ser substituídos sem prejuízo no andamento das tarefas. A esteira é a rede de comunicação de dados para a transferência dos artefatos

² ARAÚJO, L. C. G. Organização, sistemas e métodos e as tecnologias de gestão organizacional: arquitetura organizacional, benchmarking, empowerment, gestão pela qualidade total, reengenharia. 3. ed. **São Paulo: Atlas**, 2007.

ou componentes produzidos, e o andamento do trabalho e das tarefas pode ser acompanhado através de um software de monitoramento, de relatórios de tempo despendido e de produtividade.

Os ambientes virtuais, nos quais se processa todo o ciclo de desenvolvimento de um software, ficam hospedados em “servidores de desenvolvimento”. Nestes ambientes, os artefatos são desenvolvidos, corrigidos, testados e submetidos à aprovação do cliente.

2.3 EVOLUÇÃO DA FÁBRICA DE SOFTWARE POR FASES

Swanson *et al.*³ (1991 apud TENÓRIO; VALLE, 2013) apresentam a evolução das fábricas de software através de cinco fases, onde na primeira fase ocorre a organização básica e gerencia da estrutura, sendo estabelecidos os primeiros objetivos e a coleta de dados sobre o processo. Na segunda fase surge a padronização, sendo estabelecidos os primeiros métodos padronizados e introduzidas as primeiras ferramentas de desenvolvimento e bibliotecas de códigos-fonte. Na terceira fase ocorre a mecanização e o suporte ao processo, sendo introduzidas ferramentas para o controle de projetos, geração de código, teste, documentação e banco de dados. Na quarta fase ocorre um refinamento dos processos, onde é realizada a revisão dos padrões e são introduzidos novos métodos, ferramentas e o controle da qualidade, Por fim, na quinta fase são introduzidas técnicas de reutilização de componentes, design e análise de requisitos. Esta última fase se refere à fase atual das fábricas de software. O Quadro 1 apresenta a evolução por fases.

³ SWANSON, K. et al. Application Software Development: Applying Total Quality Techniques to System Development. **MIS Quarterly**, Minneapolis, Dec. 1991.

Quadro 1 - Evolução das fábricas de software por fases

Fase 1	Organização básica e gerência da estrutura (meados de 1960 e início de 1970) <ul style="list-style-type: none"> • Objetivos da manufatura de software são estabelecidos; • Foco no produto é determinado; • Começa a coleta de dados sobre o processo.
Fase 2	Customização da tecnologia e padronização (início de 1970) <ul style="list-style-type: none"> • Objetivos dos sistemas de controle são estabelecidos; • Métodos-padrão são estabelecidos para o desenvolvimento; • Desenvolvimento em ambiente on-line; • Treinamento de empregados para padronizar as habilidades; • Bibliotecas de código-fonte são introduzidas; • Começam a ser introduzidas metodologias integradas e ferramentas de desenvolvimento.
Fase 3	Mecanização e suporte ao processo (final dos anos 1970) <ul style="list-style-type: none"> • Introdução de ferramentas para apoio ao controle de projetos; • Introdução de ferramentas para a geração de código, teste e documentação; • Integração de ferramentas com banco de dados e plataformas de desenvolvimento.
Fase 4	Refinamento do processo e extensão <ul style="list-style-type: none"> • Revisão dos padrões; • Introdução de novos métodos e ferramentas; • Estabelecimento de controle e círculos da qualidade; • Transferência de métodos e ferramentas para subsidiárias e terceiros.
Fase 5	Automação flexível <ul style="list-style-type: none"> • Aumento da capacidade das ferramentas existentes; • Introdução de ferramentas de apoio à reutilização; • Introdução de ferramentas de automação de design; • Introdução de ferramentas de apoio à análise de requisitos; • Integração de ferramentas em plataformas de desenvolvimento.

Fonte: Adaptado de Swanson *et al.*⁴ (1991 apud FERNANDES; TEIXEIRA, 2004).

2.4 ATRIBUTOS DA FÁBRICA DE SOFTWARE

Para Fernandes e Teixeira (2004, p. 116), as fábricas de software requerem os seguintes atributos:

- Processos definidos e padronizados;
- Padronização das solicitações de serviço;
- Interação com o cliente, nas entradas e saídas da fábrica;
- Controle e rastreabilidade das demandas permanentes e em tempo real;
- Estimativa de custos e prazos baseados na real capacidade de atendimento;
- Tempos padrões de atendimento, estabelecidos de acordo com o domínio da aplicação, da plataforma e do tamanho da demanda;
- Rigoroso controle dos recursos em termos de alocação, disponibilidade, necessidade futura e produtividade;
- Controle e armazenamento dos artefatos e do conhecimento produzido em

⁴ SWANSON, K. et al. Application Software Development: Applying Total Quality Techniques to System Development. **MIS Quarterly**, Minneapolis, Dec. 1991.

cada etapa em bibliotecas de itens de software;

- Produtos gerados de acordo com padrões estabelecidos pela organização;
- Processos distintos para demandas de natureza diferentes;
- Equipe treinada e capacitada nos processos organizacionais e produtivos;
- Controle da qualidade do produto;
- Mecanismos de apuração e controle de custos;
- Melhoria contínua dos processos;
- Processo de atendimento ao cliente;
- Métricas e controle dos acordos de nível de serviço definidos com o cliente.

Para Tenório e Valle (2013), o controle e rastreabilidade das demandas em possibilita que o gerente de projetos saiba como está o andamento de cada demanda. Para isso, o processo de desenvolvimento deve possuir uma forma de contabilizar o tempo dedicado à cada tarefa. A estimativa de custos, prazos e tempo de atendimento devem ser baseadas na capacidade da equipe, deve haver um rigoroso controle dos recursos, em termos de alocação, disponibilidade, necessidade futura e produtividade.

Ainda segundo os autores, o controle e o armazenamento dos artefatos deve ser realizado através de bibliotecas de itens de software e o conhecimento gerado em cada etapa do desenvolvimento deve estar contido em uma base de dados, que pode ser um sistema interno ou uma base de dados adquirida. Nesta base devem constar:

- Conhecimentos relacionados à tecnologias, instalação e configuração de softwares e hardwares;
- Conhecimentos em relação à solução de problemas diversos;
- Conhecimentos inerentes ao produto de software;
- Conhecimentos gerados em cada etapa do desenvolvimento;
- A documentação integral da versão do software em produção, justificativas de adoção ou descarte de soluções, e conhecimentos necessários para a manutenção do mesmo.

Caso a documentação não esteja completa e atualizada, a produção da fábrica pode estar comprometida, pois quando o conhecimento for requerido e quem o gerou não estiver disponível, a equipe terá dificuldades ou não conseguirá seguir no desenvolvimento. Além disso, deve existir um processo definido e padronizado para o controle da qualidade dos produtos desenvolvidos.

2.5 MATURIDADE DA FÁBRICA DE SOFTWARE

Processos definidos e padronizados são uma espécie de guia seguido pela fábrica de software em seu funcionamento, desde a análise de negócio até a entrega dos produtos ao cliente. Para medir a maturidade de uma fábrica de software, Fernandes e Teixeira (2004, p. 73) trazem uma comparação entre processos maduros e processos imaturos de desenvolvimento, conforme cada medida utilizada para a medição. Este comparativo é apresentado no Quadro 2.

Quadro 2 - Comparativo entre um processo maduro e um processo imaturo

Medida	Processo Imaturo	Processo Maduro
Papéis e responsabilidades	Não são bem definidos. Cada pessoa assume seu papel, o que traz problemas à definição de responsabilidades e conflitos.	Bem definidos, com metas e medições. Relacionamentos e responsabilidades são claramente definidos.
Tratamento de mudanças	Pessoas diferentes trabalham de forma diferente e inventam sua própria forma de trabalhar.	Pessoas seguem um processo planejado, compartilham e aprendem com a experiência e seguem um processo de forma consistente.
Reação e problemas	O caos reina, apagar incêndios é normal e todos reclamam para si atos de heroísmo.	Problemas são analisados e tratados com base em conhecimento do processo; profissionalismo é a regra.
Confiabilidade	Estimativas não realistas; geralmente, o prazo de entrega e o orçamento são ultrapassados.	Estimativas são exatas e o escopo do projeto é controlado e gerenciado; metas são atingidas de forma consistente.
Recompensas às equipes	Recompensas vão para os que apagam o incêndio; se você faz certo da primeira vez, você é invisível, e este é seu trabalho; faça errado e conserte mais tarde, e você será um herói.	Recompensas vão para as equipes que produzem produtos de alta qualidade, que satisfazem aos requisitos com quase nenhuma falha; prevenção de incêndios é recompensada, ao contrário de apagar incêndios.
Previsibilidade	Você nunca sabe ao certo como está indo o desenvolvimento ou o que pode dar errado; a qualidade é variável e depende de pessoas; cronogramas e orçamentos não são baseados na experiência passada.	O progresso do projeto é previsível, assim como a qualidade dos produtos; cronogramas e orçamentos são baseados no desempenho passado e são realistas.

Fonte: Fernandes e Teixeira (2004, p. 73)

Ao analisar as colocações dos autores, um processo maduro de desenvolvimento pode ser descrito como um processo padronizado e seguido de forma consistente, onde os papéis e responsabilidades estão bem definidos. Além disso, cronogramas e orçamentos são realistas, o corpo dos projetos é gerenciado, metas são atingidas e recompensas são dadas para equipes que produzem produtos de qualidade.

2.6 TRABALHOS CORRELATOS

Nesta seção são apresentados trabalhos relacionados ao presente estudo, trazendo as propostas iniciais, as melhorias sugeridas, os resultados e as conclusões dos autores em relação aos objetivos iniciais. Também é feita uma análise entre os trabalhos correlatos e o presente estudo, para identificar pontos em comum.

2.6.1 Definição e estabelecimento de processos de fábrica de software em uma organização de TI do setor público

Em sua tese de Doutorado em Engenharia, Nomura (2008) propôs analisar, definir e estabelecer processos padronizados de Fábrica de Software, considerando a complexidade que envolve empresas do setor público, cujos pontos críticos são a descontinuidade administrativa, que é provocada pelas eleições e troca de pessoas, as constantes mudanças organizacionais, o fato de que estas empresas não visam a lucratividade, a ausência de indicadores de desempenho, os conflitos entre o pessoal permanente e o não permanente, os projetos de curto prazo, a possível descontinuidade nos projetos, a questão de projetos duplicados, entre outros. A empresa pública foco da pesquisa realizada pela autora, foi definida como EmpresaTI.

Através de estudos e de uma pesquisa-ação, foi realizado um mapeamento e a definição dos processos de desenvolvimento da empresa, considerando as características de Fábrica de Software. Após o mapeamento do processo, a autora criou a Metodologia de Desenvolvimento de Sistemas e Integração (MDSI), que levou em conta boas práticas como padronização de processos, componentes reutilizáveis, gestão de operações, terceirização e modelos de melhores práticas.

A metodologia foi implementada na empresa e segue sendo melhorada de forma contínua. A única ressalva que a autora faz, é que a implementação não ocorreu de forma corporativa, em razão das constantes mudanças organizacionais e a inexistência de áreas e grupos de especialistas dedicados ao propósito de estabelecer, acompanhar e melhorar os processos de forma contínua. Sendo assim, a implementação ocorreu em pontos específicos do desenvolvimento de software da empresa, porém com sucesso.

Relacionando o trabalho implementado pela autora com a proposta do presente trabalho, pode-se verificar que a padronização de processos e a implementação de indicadores de desempenho são uma prática constante na implementação de conceitos e boas práticas de fábrica de software.

2.6.2 Um modelo de Fábrica de Software em instituições de ensino superior

Em sua tese de Mestrado em Engenharia de Produção, Romanha (2016), teve como objetivo elaborar um plano de implementação e gerenciamento de Fábrica de Software para Instituições de Ensino Superior (IES), mais especificamente para a Associação Educacional Dom Bosco (AEDB). Para a elaboração do plano, o autor pretendia analisar os aspectos positivos e as dificuldades do processo anterior, para depois identificar as ações a serem implementadas, visando o atendimento às demandas internas da instituição.

Através de uma pesquisa documental e de questionários, Romanha realizou um estudo com algumas instituições, realizando um comparativo entre elas. Baseado nas informações obtidas, o autor elaborou sua proposta de implementação, considerando os aspectos relevantes e as expectativas da instituição.

A implementação da Fábrica de Software Acadêmica (FSA) foi efetuada com sucesso e vem atendendo satisfatoriamente às expectativas da instituição, possibilitando a participação de um maior número de alunos e professores. A implementação demandou alterações na estrutura curricular de algumas das disciplinas do curso de Sistemas de Informação, que passaram a abordar temas mais realistas, seguindo metodologias e boas práticas do mercado atual. Além disso,

segundo o autor, o trabalho contribuiu para um melhor reconhecimento da qualidade dos cursos de TI da instituição.

Relacionando o trabalho desenvolvido por Romanha com a proposta do presente trabalho, pode-se verificar que a implementação de conceitos e boas práticas de fábrica de software trazem grandes vantagens às organizações. Neste caso, a implementação da fábrica de software melhorou e padronizou o processo de desenvolvimento, que é um dos objetivos do presente trabalho.

2.7 CONSIDERAÇÕES FINAIS

Este capítulo apresentou o referencial teórico sobre “Fábrica de Software”, levando em conta o surgimento da expressão e a definição do conceito através de autores consagrados no assunto.

Além disso, foi realizado um comparativo entre fábricas tradicionais e fábricas de software, que possibilitou um melhor entendimento entre os dois modelos de fábrica, através das diferenças e pontos em comum.

Características importantes de fábrica de software também foram abordadas, como a organização, os atributos, a evolução e a maturidade. Estas características foram utilizadas posteriormente na análise e na proposta de melhoria do processo atual de desenvolvimento da organização utilizada como estudo deste trabalho.

O capítulo também apresentou trabalhos correlatos relacionados à implantação de conceitos e boas práticas de fábrica de software, que foram utilizados na análise da proposta das melhorias implantadas posteriormente.

3 PROCESSOS E METODOLOGIAS UTILIZADOS POR UMA FÁBRICA DE SOFTWARE

Para Humphrey⁵ (1989 apud TENÓRIO; VALLE, 2013), um processo de desenvolvimento de software pode ser entendido como um conjunto de passos (atividades) necessários para transformar os requisitos do usuário em software.

Segundo Swanson *et al.*⁶ (1991 apud TENÓRIO; VALLE, 2013), os processos de desenvolvimento de software possuem agregação de valor, quando aplicados preceitos de engenharia como métodos e ferramentas padronizados, apoio automatizado para o desenvolvimento, planejamento disciplinado, análise e controle de processos, e códigos e componentes reutilizáveis.

Atualmente, o principal desafio que as fábricas de software enfrentam, é desenvolver um produto que atenda às necessidades de seus clientes, com qualidade, atendimento dos prazos de entrega, baixo custo de desenvolvimento e manutenção, além de alta produtividade. Para tal, é fundamental que as fábricas de software adotem técnicas estruturadas e um processo de desenvolvimento padronizado.

A definição deste processo, que deve ser genérico o suficiente para atender em casos diversos, além de proporcionar economia de tempo e esforço, deve estabelecer uma estrutura com a padronização de metodologias de desenvolvimento, a documentação e a reutilização de componentes, a especialização dos colaboradores, e os controles da qualidade e da produtividade. Além disso, o processo deve garantir que os colaboradores possam ser substituídos sem prejuízo no processo produtivo, devem estar definidas as tarefas e os produtos (artefatos) gerados, bem como os responsáveis pela execução de cada uma das etapas.

Assim como uma eventual definição de processo de desenvolvimento, a alteração de um processo em uma determinada organização é uma tarefa árdua e arriscada, demanda tempo para apresentar resultados e pode mudar radicalmente o modo como as pessoas trabalham ou veem seu trabalho. Esta mudança requer planejamento, monitoramento, gerenciamento e avaliação constantes. Geralmente é

⁵ HUMPHREY, W. *Managing the Software Process*. Reading, MA: Addison-Wesley, 1989.

⁶ SWANSON, K. et al. *Application Software Development: Applying Total Quality Techniques to System Development*. MIS Quarterly, Minneapolis, Dec. 1991.

muito mais marcante do que a substituição de ferramentas ou tecnologias, que muitas vezes não alteram o processo.

Este capítulo apresenta um breve histórico do surgimento e evolução dos processos e metodologias de desenvolvimento em fábricas de software. Em seguida, são apresentados o estudo de aprimoramento do processo de desenvolvimento de software, processos tradicionais, processos ágeis e modelos e processos de maturidade. Ao final do capítulo são apresentadas as considerações finais.

3.1 HISTÓRICO

No início dos anos 60, a área de desenvolvimento não era vista como uma área que necessitava ser gerenciada. A qualidade somente estava presente somente na inspeção do produto, para identificar, corrigir defeitos e evitar falhas na operação. Entre as décadas de 60 e 80, eram os grandes fabricantes de hardware e software que forneciam os principais modelos de processos. Por este motivo, era utilizado o termo “processos proprietários”.

Fernandes e Teixeira (2004, p. 23), ilustram a evolução do desenvolvimento de software ao longo do tempo, trazendo as operações, os processos, as plataformas e as metodologias de cada dez anos, iniciando pelo ano de 1960 até o século atual. O Quadro 3 ilustra a evolução do desenvolvimento de software.

Quadro 3 - Evolução do desenvolvimento de software

	1960-1970	1970-1980	1980-1990	1990-2000	Século XXI
Operações	Artesanal	Artesanal	Fábrica SW	Fábrica SW Integrated Outsourcing	SPL
Processos	Processos proprietários		CMM	PMI, ASAP	XP, ASD, LD
Plataformas	Fortran, Assembler	Cobol, PL1	Natural C, C++, Clipper	VB, Delphi, Oracle	Java, .NET, XML
Metodologias	Waterfall	Estruturada, Essencial	Estruturada, Essencial	OO, UML, Componentes	?

Fonte: Fernandes e Teixeira (2004, p. 23)

Posteriormente, através de um forte movimento do governo americano, buscou-se introduzir na gestão do software, práticas da Gestão da Qualidade Total (Total Quality Management), fazendo com que surgisse o modelo CMM (Capability Maturity Model).

Um grande avanço ocorreu também a partir das metodologias de análise estruturada, essencial, orientada a objetos e ágeis, pois foi possível escrever programas de forma mais clara, ágil e fácil de modificar. Com a aplicação do conceito de fábrica de software e os resultados positivos obtidos, os processos padronizados de desenvolvimento foram difundidos pelo mundo, sendo inclusive utilizados em fábricas dispersas geograficamente.

3.2 APRIMORAMENTO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Sommerville (2007) afirma que há uma estreita relação entre a qualidade de um processo de desenvolvimento e a qualidade dos produtos desenvolvidos por meio deste processo, e que devido a esse fato muitas empresas buscam o aprimoramento de seus processos de desenvolvimento.

O autor afirma ainda que o aprimoramento de processos não significa simplesmente adotar métodos específicos, ferramentas ou algum modelo de processo padronizado. Existem outros fatores organizacionais, culturais que influenciam no processo de desenvolvimento de cada organização.

Na concepção do autor, o aprimoramento do processo compreende os estágios de medição, de análise e de mudança.

3.2.1 Medir

Corresponde ao estágio de medição do processo, onde são medidos os atributos do atual projeto ou produto. O objetivo do estágio é aprimorar as medidas de acordo com os objetivos da organização envolvida.

Neste estágio, as medições do processo são realizadas através de dados quantitativos e em três classes de métricas: tempo total de execução do processo, recursos necessários para a execução e o número de ocorrências de um evento específico.

Sommerville (2007) afirma que a dificuldade da medição é saber o que medir e que o aprimoramento do processo deve ser guiado por objetivos, questões e métricas, sugerindo a abordagem “Objetivo-Questão-Métrica” (GQM). A abordagem possui as seguintes características:

- **Objetivos:** Correspondem ao que a organização busca e devem estar alinhados ao negócio. São os objetivos que orientam as mudanças no atual processo;
- **Questões:** Correspondem aos refinamentos dos objetivos, em que são identificadas as áreas de incertezas. Cada objetivo pode ter uma ou mais questões, que são as perguntas a serem respondidas para validar se o objetivo foi alcançado;
- **Métricas:** Correspondem às medições a serem coletadas para responder às perguntas (questões) e confirmar se o objetivo foi alcançado.

A vantagem em aplicar a abordagem no aprimoramento de processos, é que ela faz a separação de assuntos organizacionais como os objetivos, de processos específicos como as questões.

3.2.2 Analisar

Corresponde ao estágio de análise, onde o processo atual é analisado e medido, bem como seus pontos fortes e seus pontos fracos. Após a análise, é gerado o modelo do processo atual, com as características principais e a identificação das mudanças a serem implementadas.

Sommerville (2007) afirma que o estágio inicial de análise é qualitativo, onde o analista tenta descobrir as características do modelo. Os estágios posteriores podem ser quantitativos e fazem uso das medições de processo coletadas. As técnicas de análise sugeridas pelo autor são questionários, entrevistas, observação e

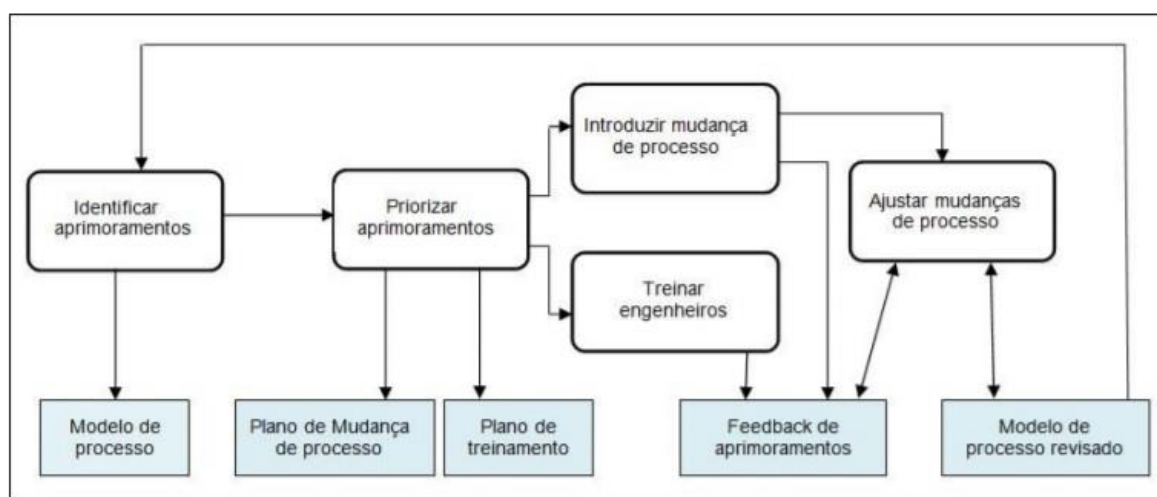
estudos etnográficos. As três primeiras técnicas, são utilizadas neste trabalho para o estágio de análise do processo atual.

3.2.3 Mudar

Corresponde ao estágio de mudança do processo, onde são implementadas as mudanças identificadas no estágio de análise. Essas mudanças são guiadas pelos objetivos definidos na fase de medição, e podem ocorrer através de novas práticas, métodos ou ferramentas, alteração de ordem das atividades do processo, introdução ou remoção de entregas e introdução de novos papéis e responsabilidades.

Conforme Sommerville (2007), os cinco estágios do processo de mudança são identificar aprimoramentos, priorizar aprimoramentos, introduzir mudança de processo, treinar a mudança de processo e ajustar as mudanças de processo. A Figura 1 mostra o processo de mudança sugerido por Sommerville (2007).

Figura 1 - Processo de mudança do processo



Fonte: Sommerville (2007)

A identificação de aprimoramentos consiste em identificar os pontos a serem melhorados no processo para atingir os objetivos da mudança, com base na análise dos resultados obtidos no estágio da análise do processo.

A priorização dos aprimoramentos consiste em avaliar as possíveis mudanças e priorizá-las para serem implementadas. Esta atividade possui ainda mais

importância quando há várias mudanças propostas e as mesmas não poderão ser implementadas ao mesmo tempo.

A introdução da mudança de processo consiste em colocar em prática os novos procedimentos, métodos e ferramentas, a fim de integrá-los com as demais atividades do processo.

O treinamento das mudanças é para que a equipe tenha o entendimento e a clareza no impacto de suas atividades, gerando melhorias e aceitando as mudanças.

Por fim, na fase de ajuste das mudanças ocorre o acompanhamento das mudanças introduzidas e as modificações necessárias quando descobertos problemas no novo processo.

3.3 PROCESSOS TRADICIONAIS

Nos processos tradicionais de desenvolvimento o foco está no levantamento e no detalhamento rigoroso de requisitos, onde são definidas e documentadas todas as necessidades do cliente. Como o tipo é caracterizado pelo detalhamento no planejamento, os processos de análise podem ser demorados e de difícil manutenção.

3.3.1 Método Cascata (Waterfall)

O modelo Cascata sugere uma abordagem sequencial para o desenvolvimento de software, iniciando pela especificação dos requisitos, seguido pelo planejamento, pela modelagem, pela construção e pela disponibilização.

A utilização do método é ideal para sistemas ou softwares em que os requisitos são bem compreendidos e não há mudanças constantes, ou seja, os requisitos são estáveis e ocorrem apenas pequenas melhorias ou aperfeiçoamentos em um sistema ou software já existente.

O método cascata possui também uma variação denominada modelo V, que descreve a relação entre ações de comunicação, modelagem e atividade de construção, com ações de garantia de qualidade. À medida que a equipe executa as

ações relacionadas à parte esquerda do modelo, os requisitos do problema vão sendo refinados em representações cada vez mais detalhadas. Após a geração do código, a equipe passa para o lado direito do modelo, realizando uma série de testes para validar o que foi criado no lado esquerdo.

Se comparados o ciclo de vida do método cascata e do modelo V, não há diferenças. O modelo V apenas oferece uma forma de visualizar como as ações de validação e verificação são aplicadas.

A aplicação do método cascata, pode culminar nos seguintes problemas:

- Os projetos raramente seguem um fluxo sequencial;
- O modelo necessita que o cliente estabeleça suas necessidades de forma precisa, algo que não ocorre frequentemente;
- Uma versão operacional do programa somente estará disponível quando a equipe estiver próxima do final do projeto, fato que pode gerar problemas com o cliente;
- Um erro grave pode ser desastroso, se não detectado rapidamente.

3.4 PROCESSOS ÁGEIS

Nos processos ágeis de desenvolvimento o foco é a eficiência, a versatilidade e o ciclo de vida curto, portanto não possui tanto rigor quanto o processo tradicional. Além disso, os processos ágeis pressupõem que a análise de riscos seja bastante flexível, tendo em vista a constante mudança das necessidades do cliente. Não há uma etapa definida para os replanejamentos, pois estes são constantes durante todo o processo.

3.4.1 Software Product Lines

O SPL é uma operação que utiliza uma mesma arquitetura para o desenvolvimento de produtos que estão relacionados uns com os outros, utilizando componentes reutilizáveis. Desta forma, os produtos possuem funcionalidades em comum e funcionalidades próprias.

A reutilização dos componentes nestes produtos, faz com que ocorram vantagens econômicas no desenvolvimento, através da redução do custo, da redução do tempo de desenvolvimento e manutenção, do aumento da qualidade e da estimativa do custo, que pode ser mensurada de forma mais precisa. Há também possíveis riscos em utilizar a operação, como um investimento alto no desenvolvimento de produtos que podem sofrer grandes alterações em seus principais requisitos, fazendo com que os produtos não estejam mais relacionados.

3.4.2 Accelerated SAP

A ASAP aplica ferramentas para a geração de resultados confiáveis, aumentando assim as chances de sucesso das empresas que aplicam o método. A metodologia é baseada em cinco pilares, conforme as seguintes fases:

- **Preparação do projeto:** Nesta fase são estabelecidos os usuários chave, os colaboradores envolvidos, o planejamento e o desenho do escopo do projeto, as estratégias adotadas, a definição e o treinamento das equipes, a organização dos documentos necessários, aspectos técnicos, a identificação dos riscos e o controle da qualidade;
- **Modelo dos processos:** Fase onde é gerado um documento nomeado "Business BluePrint". Através do mesmo, o gestor tem acesso ao detalhamento dos processos e das exigências referentes ao negócio da empresa, como a estruturação da organização, o desenho e revisão dos processos, a gestão do projeto, a documentação, as mudanças, a definição das estratégias e o estabelecimento de novos processos;
- **Realização:** Fase em que é colocado em prática o documento gerado na etapa de modelo dos processos. Além disso, ocorrem etapas como a realização de testes, a definição dos parâmetros do sistema e o desenvolvimento e a customização de programas;
- **Preparação final:** Nesta fase são realizadas as últimas verificações, como a elaboração da implementação, os testes de volume de dados, o treinamento dos usuários finais e o controle da qualidade;

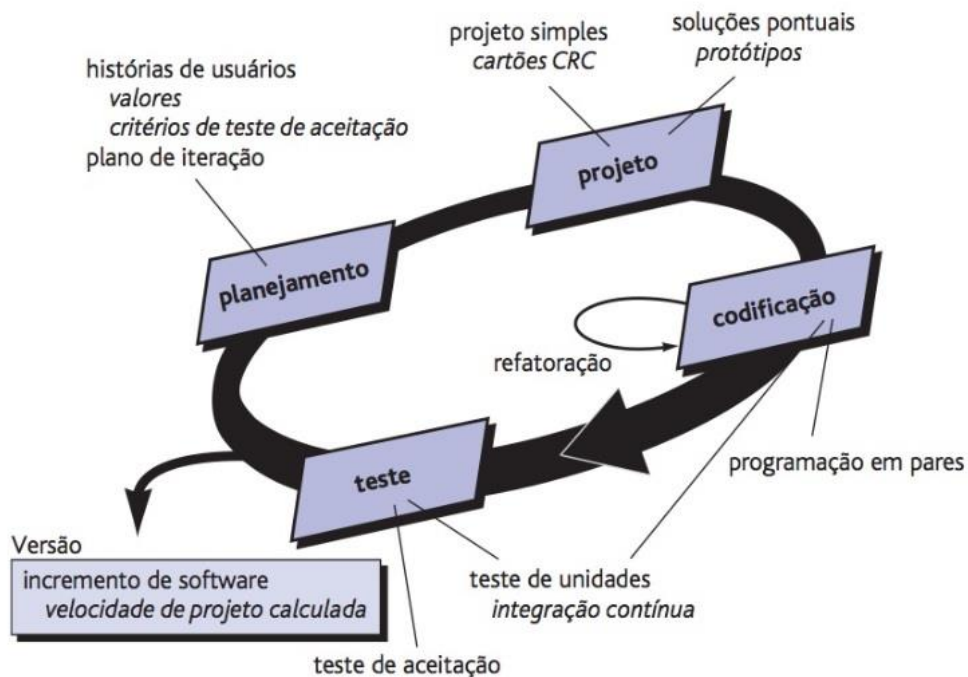
- **“Go Live” e suporte:** Fase onde o sistema é colocado em prática e os processos ocorrem com melhorias de performance. Além disso, ocorrem atividades como os ajustes no software, na base de dados e no sistema operacional e o suporte aos usuários.

A implementação da metodologia ASAP, proporciona às empresas vantagens como o fornecimento de melhores diretrizes, uma melhor preparação para eventuais mudanças, a possibilidade de que todas as áreas da organização trabalhem de forma holística, a melhoria dos processos atuais, a minimização de erros e a redução de custos, fazendo com que a empresa que a implementa tenha maior poder de competitividade no mercado.

3.4.3 Extreme Programming

Em português “Programação Extrema”, o XP é uma metodologia ágil orientada a objetos, que atualmente é a abordagem mais utilizada para o desenvolvimento de software ágil. A Figura 2 apresenta o processo da XP.

Figura 2 - Processo da XP



Planejamento

A etapa inicia com o levantamento de requisitos. De forma a entender os negócios do software e obter uma ampla percepção sobre os resultados solicitados pelo cliente, os fatores principais e as funcionalidades, é criado um conjunto de histórias. Cada história (que é similar aos casos de uso) é escrita pelo cliente e colocada em uma ficha. Posteriormente, o cliente atribui um valor às histórias, baseando-se no valor de negócio dos recursos ou funções. Os membros da equipe avaliam cada uma das histórias escritas, atribuindo um custo que é medido em semanas de desenvolvimento. Se a história exigir mais do que três semanas de desenvolvimento, a mesma é dividida em histórias menores, ocorrendo novamente a atribuição de valores e custos. Nesta etapas, novas histórias podem ser escritas a qualquer momento.

Para definir o próximo incremento a ser desenvolvido pela equipe XP, os clientes e desenvolvedores trabalham juntos para decidir como agrupar as histórias para a versão seguinte. Após os mesmos estarem de acordo sobre as histórias incluídas, datas de entrega e outras questões, a equipe ordena as histórias a serem desenvolvidas em uma das três formas:

- Todas serão implementadas imediatamente;
- As histórias de maior valor serão deslocadas para cima no cronograma e implementadas primeiro;
- As histórias de maior risco serão deslocadas para cima no cronograma e implementadas primeiro.

Após a entrega da primeira versão do projeto, a equipe XP calcula a velocidade do projeto, que é o número de histórias de clientes implementadas durante a primeira versão. A velocidade é utilizada para estimar as datas de entrega e o cronograma para as versões subsequentes, e para determinar se foi assumido um compromisso exagerado para todas as histórias ao longo do projeto. Se ocorrer um exagero, são alterados ou o conteúdo das versões ou as datas de entrega.

Conforme o andamento do desenvolvimento, o cliente pode acrescentar, mudar os valores, fazer a divisão ou eliminar histórias já existentes.

Projeto

O projeto XP segue os princípios da simplificação, e estimula o uso de cartões CRC (Class Responsibility Collaboration Card), cartões de colaboração de responsabilidade de classe, uma ferramenta de brainstorming utilizada no design de software orientado a objetos. Os cartões CRC são o único artefato produzido como parte do processo XP.

Se encontrado um problema de projeto difícil como parte de uma história, é recomendado criar um protótipo operacional para esta parte do projeto, denominada solução pontual. Este protótipo é implementado e avaliado, como o objetivo de reduzir o risco para quando a verdadeira implementação iniciar, além de validar as estimativas originais para a história.

A XP estimula a refatoração, pois o projeto deve ser continuamente modificado conforme a construção segue. O objetivo da refatoração é controlar essas modificações, sugerindo pequenas mudanças no projeto, a fim de melhorá-lo. Outro aspecto importante da XP, é que a elaboração do projeto ocorre tanto antes quanto depois de iniciada a codificação.

Codificação

Após desenvolvidas as histórias e efetuada a elaboração do projeto, ao invés de codificar, a equipe desenvolve uma série de testes unitários, relacionados à cada uma das histórias incluídas na versão a ser incrementada. Uma vez criados tais testes, a equipe se concentra na codificação, para que depois o código seja testado, fornecendo o feedback imediato.

Um conceito recomendado pela XP, é que ocorra a programação em pares, ou seja, duas pessoas trabalhando juntas em uma determinada história. Tal prática fornece um mecanismo para a solução de problemas e garantia da qualidade em tempo real, pois o código é revisto à medida que é criado.

Quando a dupla concluiu o trabalho, é realizada a integração com os demais trabalhos desenvolvidos. A estratégia da integração contínua, evita problemas de compatibilidade e interface, além de revelar erros precocemente.

Testes

A etapa de testes da XP contempla os testes de unidades e os testes de aceitação. Os testes de unidades criados devem ser implementados através de uma metodologia automatizada, para que possam ser executados fácil e repetidamente, já que o código é frequentemente modificado. Com esta metodologia criada, os testes de integração e validação do sistema podem ocorrer diariamente, fazendo com que os problemas sejam corrigidos logo no início da codificação.

Os testes de aceitação são os testes especificados pelo cliente, e mantêm o foco nas características e funcionalidades do sistema. Estes testes são obtidos das histórias dos usuários, implementadas como parte de uma versão do software.

3.4.4 Adaptive Software Development

A ASD é uma metodologia de desenvolvimento de software criada por Jim Highsmith, que dá grande importância às pessoas e aos resultados, além da máxima colaboração entre a equipe de desenvolvimento e o cliente. A metodologia é direcionada ao desenvolvimento ágil e com alto grau de mudanças, e prioriza o entendimento sobre a documentação, a colaboração sobre o controle e a adaptação sobre a otimização. A mesma segue o ciclo de vida dinâmico Adaptive, que compreende as fases Especulação, Colaboração e Aprendizagem, ao invés do ciclo de vida estático (Planejar, Projetar e Construir). Este ciclo de vida é dedicado a aprendizagem contínua e direcionado a aceitar mudanças constantes, a executar reavaliações e estimular a intensa colaboração entre desenvolvedores, testadores e clientes. A Figura 3 apresenta o ciclo de vida Adaptive.

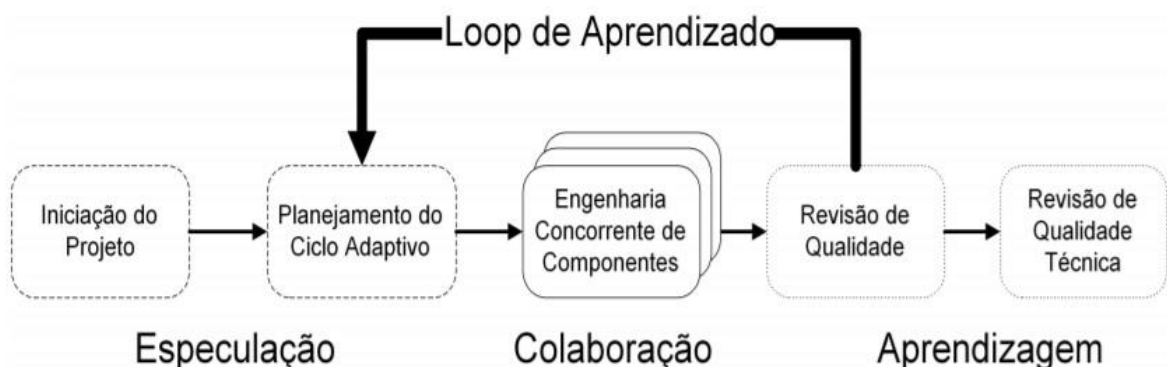
Figura 3 - Ciclo de vida Adaptive



Fonte: Highsmith⁷ (2000 apud NASCIMENTO, 2008)

O ciclo possui características como foco na missão, desenvolvimento baseado em componentes, iterações, timebox, análise de riscos e tolerância à mudanças. Estas características direcionam o processo de desenvolvimento, com o planejamento e os resultados de cada iteração, a análise de riscos e a previsão de mudanças para o próximo ciclo. As fases podem ser divididas em porções menores, possibilitando um melhor entendimento do ciclo e atividades envolvidas, conforme mostra a Figura 4.

Figura 4 - Loop de aprendizado



Fonte: Highsmith (2000 apud NASCIMENTO, 2008)

⁷ HIGHSMITH, J. Lifecycle Dinosaurs. Using Adaptive Software Development to meet the challenges of a high-speed, high-change environment. **Software Testing and Quality Engineering**, 2000.

Especulação

Esta fase inicia pela atividade de iniciação, que determina os objetivos do projeto, através do entendimento e documentação de restrições, do estabelecimento da organização e responsáveis, da identificação e do esboço dos requisitos, da estimativa do escopo e tamanho do projeto, e da identificação dos principais riscos. Baseado nos resultados obtidos, são definidas as funcionalidades que serão desenvolvidas.

Após a atividade de iniciação, é determinado o número total de ciclos, associando um timebox para cada um deles. O tamanho de ciclos é determinado pelo cronograma geral e pelo grau de incerteza do projeto. Para a identificação de defeitos e falhas, são criados marcos para cada ciclo.

Os passos finais da fase efetuam a associação dos componentes aos ciclos e a definição das tarefas a serem realizadas, sendo entregues produtos tangíveis ao final de cada um dos ciclos.

Colaboração

Esta fase é composta pela etapa Engenharia Concorrente de Componentes, que trata de assuntos envolvendo os relacionamentos entre os membros da equipe. Como as distintas opiniões devem ser gerenciadas da melhor forma possível, a fase sugere técnicas como a programação em pares e a propriedade coletiva do código, minimizando assim estes problemas.

Aprendizagem

Esta fase é composta pelas etapas Revisão de qualidade e Revisão de qualidade técnica. Nesta fase, a aprendizagem é o resultado da aplicação de técnicas de revisão de qualidade, aplicadas ao final de cada ciclo de desenvolvimento. Para a realização desta revisão, devem ser observadas a qualidade do produto na visão técnica e do usuário final, além da performance e das práticas da equipe responsável pelas entregas. Estes retornos são fundamentais para o sucesso do projeto.

3.4.5 Lean Development

A LD é uma metodologia que surgiu do movimento Lean de produção das fábricas nos anos 80. Diferentemente dos métodos ágeis, a LD é mais focada na estratégia e seu principal objetivo é a redução de custos, tempo e defeitos.

Womack e Jones⁸ (2003 apud BENZECRY, 2017), afirma que o pensamento Lean é um forte antídoto ao desperdício, realizando cada vez mais com menos, menos esforço humano, menos equipamentos, menos tempo e menos espaço, enquanto se aproxima dos clientes para saber exatamente o que eles necessitam. Provendo um imediato feedback, o pensamento Lean transforma desperdício em valor.

Ainda segundo Ribeiro e Ribeiro⁹ (2015 apud BENZECRY, 2017), o Lean busca a redução do desperdício, sendo este tudo aquilo que não é realizado para o cliente. Ele também lista os seguintes conceitos para a identificação da LD:

- Eliminar o desperdício;
- Dar poder ao time (Permitir que a equipe seja responsável por decisões técnicas);
- Entregar rapidamente (Entregas rápidas e constantes geram valor para o cliente, priorizando as funcionalidades mais importantes);
- Ver o todo (Entender o sistema como um todo e não apenas uma soma de partes);
- Construir com qualidade (Qualidade testada a cada etapa);
- Adiar decisões (Espera mais informação para diminuir os riscos);
- Amplificar o aprendizado (“feedback” rápido e aprendizado contínuo).

⁸ WOMACK, James P.; JONES, Daniel T. Lean thinking: Banish waste and create wealth in your Corporation. **New York: Free Press**, 2003.

⁹ RIBEIRO, Rafael Dias; RIBEIRO, Horácio da Cunha e S. Métodos ágeis em gerenciamento de projetos. **Rio de Janeiro: Horácio da Cunha e Sousa Ribeiro**, 2015.

3.5 MODELOS E PROCESSOS DE MATURIDADE

Modelos e processos de maturidade são uma referência sobre o que é consolidado em relação às atividades que envolvem o desenvolvimento de software. Além disso, são compostos pelas melhores práticas de desenvolvimento e manutenção que cobrem o ciclo de vida dos produtos da fábrica de software, desde a concepção até a entrega e manutenção.

Um processo maduro de desenvolvimento pode ser descrito como um processo padronizado, que é melhorado e seguido de forma consistente.

3.5.1 Capability Maturity Model

O CMM aplica conceitos de gestão de processos e melhoria da qualidade, tanto para o desenvolvimento quanto para a manutenção de sistemas e softwares. Além disso, pode ser considerado um guia para a melhoria contínua e descreve o amadurecimento de uma organização ao longo do aperfeiçoamento dos processos.

O CMM é formado pelos seguintes níveis de maturidade:

- **Inicial:** Nível que representa um cenário até certo ponto caótico. Há poucos processos definidos e o sucesso da organização depende das pessoas;
- **Repetível:** Nível onde são estabelecidos procedimentos, políticas, controles básicos, processos padronizados, estimativas baseadas no entendimento dos requisitos, monitoramento de custos, cronogramas e funcionalidades, além do controle de mudanças e o rastreamento dos produtos de software;
- **Definido:** Nível onde são documentados os processos padronizados de desenvolvimento e manutenção. Além disso, ocorrem a adaptação do processo padrão conforme particularidades de cada projeto, a melhoria contínua dos processos, a implantação de um programa de treinamentos e a aplicação de técnicas de engenharia de software;
- **Gerenciado:** Nível onde são estabelecidas metas quantitativas, relativas à qualidade dos processos e dos produtos. São gerenciados os riscos e a

medição da qualidade e da produtividade é utilizada para gerenciar melhorias;

- **Otimização:** Nível onde o foco é a melhoria contínua. Mudanças nos processos e nas tecnologias precisam ser planejadas e gerenciadas, e a organização enfatiza a prevenção.

Segundo Fernandes e Teixeira (2004, p. 79), os níveis de maturidade representam as faixas de resultados esperados em um processo de software. Cada nível é uma camada de fundação para melhoria contínua, onde as áreas-chave ou KPAs, são os principais blocos para o estabelecimento das mesmas, além de identificar os grupos de atividades de cada nível de maturidade. O Quadro 4 apresenta as áreas-chave, na concepção dos autores.

Quadro 4 - Áreas-chave por nível do CMM

Nível Repetível	Nível Definido	Nível Gerenciado	Nível Otimização
Gestão dos requisitos	Organização com foco no processo	Gestão quantitativa do processo	Prevenção de defeitos
Planejamento do projeto de software	Definição do processo da organização	Gestão da qualidade do software	Gestão da mudança tecnológica
Controle do projeto de software	Programa de treinamento		Gestão da mudança no processo
Gestão do subcontrato de software	Gestão integrada do software		
Garantia da qualidade de software	Engenharia do produto de software		
Gestão de configuração	Coordenação intergrupos		
	Revisões conjuntas		

Fonte: Adaptado de Fernandes e Teixeira (2004, p. 80)

3.5.2 Capability Maturity Model Integration

O CMMI é uma evolução do conceito dos modelos estabelecidos pelo CMM e foi projetado para a melhoria dos processos de desenvolvimento de produtos e

serviços, assim como aquisições e manutenções. O modelo compreende a gestão de processos, de projetos e de fornecimento, engenharia de sistemas e de software, e o processo de desenvolvimento integrado ao produto.

O modelo possui as seguintes representações, uma contínua e a outra baseada em estágios:

a) Contínua: A representação é composta pelos seguintes níveis:

- Incompleto: O processo é parcialmente desempenhado ou não é desempenhado. As metas podem não ser satisfeitas;
- Desempenhado: É um processo que satisfaz às metas de cada uma das suas áreas, apoiando a produção dos produtos esperados;
- Gerenciado: É um processo desempenhado executado conforme políticas organizacionais. O processo é planejado, revisado, avaliado, gerenciado, controlado e institucionalizado, empregando pessoas habilitadas, alocando os recursos e envolvendo os stakeholders adequados;
- Definido: É um processo gerenciado e personalizado com base em processos padronizados, gerando ativos que podem ser reutilizados. Este processo evolui e é melhorado continuamente, além de ser disseminado por toda a organização;
- Gerenciado Quantitativamente: É um processo definido e controlado por técnicas estatísticas e métodos quantitativos. Para a gestão deste processo são estabelecidos objetivos quantitativos de qualidade e desempenho;
- Otimizado: É um processo gerenciado quantitativamente e adaptado aos objetivos de negócio. Através de aperfeiçoamentos e inovação, o processo foca na melhoria contínua e as melhorias propostas são avaliadas quantitativamente.

b) Por estágios: A representação por estágio é semelhante ao CMM e possui como estágios o inicial, o gerenciado, o definido, o gerenciado quantitativamente e o otimizado. Além dos estágios, o processo é composto

por áreas. Cada área possui metas, que podem ser específicas ou genéricas. O Quadro 5 apresenta as áreas-chave por estágio.

Quadro 5 - Áreas-chave por estágio do CMMI

Nível Gerenciado	Nível Definido	Nível Gerenciado Quantitativamente	Nível Otimizado
Gestão de requisitos	Desenvolvimento dos requerimentos	Desempenho do processo organizacional	Inovação organizacional e disseminação
Planejamento do projeto	Solução técnica	Gestão quantitativa do projeto	Análise causal e resolução
Controle e monitoramento do projeto	Integração do produto		
Gestão do acordo com o fornecedor	Verificação		
Medição e análise	Validação		
Quality assurance do processo e do produto	Foco no processo organizacional		
Gestão da configuração	Definição do processo organizacional		
	Treinamento organizacional		
	Gestão integrada do projeto		
	Gestão do risco		
	Integração da equipe		
	Gestão integrada do fornecedor		
	Análise de decisão e resolução		
	Ambiente organizacional para a integração		

Fonte: Adaptado de Fernandes e Teixeira (2004, p. 86)

O Quadro 6 apresenta a categorização das áreas dos processos.

Quadro 6 - Categorização das áreas dos processos

Processo	Área
Gestão do processo	Foco no processo organizacional Definição do processo organizacional Treinamento organizacional Desempenho do processo organizacional Inovação organizacional e disseminação
Gestão do projeto	Planejamento do projeto Controle e monitoramento do projeto Gestão do acordo com o fornecedor Gestão integrada do projeto Gestão do risco Integração da equipe Gestão integrada do fornecedor Gestão quantitativa do projeto
Engenharia	Desenvolvimento do requerimentos Gestão de requisitos Solução técnica Integração do produto Verificação Validação
Suporte	Gestão da configuração Quality assurance do projeto e do produto Medição e análise Ambiente organizacional para integração Análise de decisão e resolução Análise causal e resolução

Fonte: Adaptado de Fernandes e Teixeira (2004, p. 87)

3.6 CONSIDERAÇÕES FINAIS

Neste capítulo foram abordadas questões referentes aos processos e metodologias utilizadas por uma fábrica de software.

A fim de complementar a fundamentação teórica elaborada no Capítulo 2, foi efetuado um levantamento histórico do desenvolvimento de software, contemplando os principais processos e metodologias utilizados ao longo do tempo, como processos tradicionais, processos ágeis e modelos e processos de maturidade.

Para nortear a proposta de implementação de conceitos e boas práticas de fábrica de software, que é o objetivo deste trabalho, foi estudada a estrutura do processo de aprimoramento de um processo de desenvolvimento.

O Capítulo 4, que é apresentado a seguir, traz a medição e a análise do atual processo de desenvolvimento da organização utilizada como estudo de caso deste trabalho, bem como os objetivos, questões e métricas que serão utilizados na proposta de melhoria do processo.

4 MEDIÇÃO E ANÁLISE DO PROCESSO ATUAL

Este capítulo tem por objetivo apresentar o estudo do processo atual, assim como informações relevantes para a contextualização das características da equipe e da organização.

A equipe objeto de estudo deste trabalho é composta por dois gerentes, que atuam também como gerentes de projetos e analistas, além de duas pessoas que atuam como “Help Desk” e 32 desenvolvedores, alguns deles também atuam como analistas. A organização é o Departamento de Informática da empresa Tramontina Central de Administração.

A equipe desenvolve e dá manutenção em diversos sistemas e softwares, que são utilizados por colaboradores de todas as empresas do grupo Tramontina. Os sistemas desenvolvidos são os sistemas necessários para o funcionamento das empresas, como Planejamento e Controle da Produção (PCP), Desenvolvimento Humano (DH), Folha de Pagamento, Medicina do Trabalho, Contabilidade, Gerenciamento da Qualidade, Assistência Técnica, Mecânica, Compras, Almoxarifado, entre outros. Por se tratarem de inúmeros usuários, cada sistema possui uma comissão chamada Grupo de Trabalho (GT), formada geralmente pela pessoa que mais conhece sobre o sistema em cada unidade do grupo.

As linguagens de desenvolvimento utilizadas pelo Departamento de Informática são Genero, Delphi e alguma coisa de linguagens Web como JavaScript. O banco de dados é o Informix da IBM.

São inúmeros os usuários atendidos pelo Departamento de Informática. Os mesmos estão localizados em vários países, dado que a marca Tramontina está presente em mais de 120 países. Os usuários trabalham em empresas distintas, sendo fábricas, centros de distribuição, varejos, lojas e escritórios comerciais.

O capítulo está organizado em três seções, onde na Seção 4.1 são apresentadas as definições dos objetivos da implantação de boas práticas de Fábrica de Software, conforme a abordagem GQM (Goal-Question-Metric), sugerida por Sommerville (2007) no estágio de medição do processo.

Na Seção 4.2, são apresentados a análise e o modelo do processo atual, além das técnicas aplicadas e os resultados obtidos. Estas técnicas fazem parte do estágio de análise do processo citado, enquanto o modelo é o que deve ser gerado como resultado do estágio da análise do processo.

Por fim, na Seção 4.3 são apresentadas as considerações finais do capítulo.

4.1 MEDIÇÃO DO PROCESSO ATUAL

Segundo Sommerville (2007), o aprimoramento do processo deve ser guiado por objetivos. Esses objetivos devem ser definidos através da abordagem Objetivo-Questão-Métrica, ou GQM (Goal-Question-Metric), no estágio de medição do processo.

Para a definição dos objetivos a serem alcançados, foram analisados os principais pontos críticos do atual processo de desenvolvimento, que são a falta de um processo de desenvolvimento definido, padronizado e documentado, a alta quantidade de retrabalhos na análise de requisitos e no desenvolvimento de funcionalidades, apesar de não existir um índice de medição atualmente, e o alto índice de erros em funcionalidades entregues.

Fernandes e Teixeira (2004, p. 116) citam que um dos atributos requeridos por uma fábrica de software é a definição e padronização dos processos. Estes atributos foram estudados na seção de atributos de uma fábrica de software, através do Capítulo 2. Sobre reduzir o índice de erros na entrega das funcionalidades, essa é uma das formas de reduzir a quantidade de retrabalhos.

Sendo assim, para o estudo de caso aplicado através deste trabalho, foram definidos os seguintes objetivos:

- **Definir e documentar um processo padronizado de desenvolvimento;**
- **Deixar mais claro o processo de análise de requisitos;**
- **Aumentar a eficiência da equipe de desenvolvimento;**
- **Aumentar a satisfação do cliente.**

Para cada objetivo, assim como sugere a abordagem GQM, foram definidas as questões e métricas. As mesmas são apresentadas a seguir:

4.1.1 Definir e documentar um processo padronizado de desenvolvimento

Este objetivo é relacionado ao atual processo de desenvolvimento, visto que padronizando atividades, ferramentas, papéis e responsabilidades, a equipe terá maior clareza do que e como fazer, fato que pode fazer com que a equipe seja mais eficiente. Além disso, através do modelo formal do processo padronizado, poderá haver maior facilidade de ambientação de novos membros. Para o objetivo, foram definidas as seguintes questões e métricas:

a) Questão Q1.1: Como identificar se o processo foi padronizado e documentado?

- **Métrica M1.1:** Razão entre a quantidade de etapas do processo e a quantidade de etapas do processo que estão documentadas.

b) Questão Q1.2: Quantas demandas ou quantos projetos chegaram para o desenvolvimento, que estão seguindo o processo padronizado?

- **Métrica M1.2:** Quantidade de demandas ou projetos que seguem o padrão, através da auditoria realizada.

4.1.2 Deixar mais claro o processo de análise de requisitos

Este objetivo é relacionado à etapa de análise de requisitos, visto que deixando mais claro o processo de análise, a equipe terá maior clareza do que deve ser desenvolvido. Além disso, se entendidas com mais clareza as necessidades do cliente, a equipe automaticamente se torna mais eficiente, pois não haverá retrabalho e readequação de requisitos. Para o objetivo foi definida a seguinte questão e métricas:

a) Questão Q2.1: Como reduzir a quantidade de alterações de requisitos realizadas na etapa de análise, após a entradas das demandas?

- **Métrica M2.1:** Quantidade de alterações realizadas por requisito, depois da entrada da demanda e entrega ao cliente.
- **Métrica M2.2:** Razão entre a quantidade de requisitos modificados sobre a quantidade de requisitos solicitados.

4.1.3 Aumentar a eficiência da equipe de desenvolvimento

Este objetivo é relacionado ao atual processo e equipe de desenvolvimento, visto que identificando o que é retrabalho e reduzindo os índices de retrabalho e erros, automaticamente a equipe se torna mais eficiente e mais funcionalidades são desenvolvidas e liberadas. Para o objetivo, foram definidas as seguintes questões e métricas:

a) **Questão Q3.1:** Como identificar o que é retrabalho?

- **Métrica M3.1:** Quantidade de alterações efetuadas em demandas ou projetos entregues em N dias.

b) **Questão Q3.2:** Como reduzir o número de retrabalhos gerados?

- **Métrica M3.2:** Percentual da razão entre o número de funcionalidades de retrabalho sobre o número de funcionalidades entregues.

c) **Questão Q3.3:** Como reduzir o índice de erros de programação detectados na fase de testes?

- **Métrica M3.3:** Percentual da razão entre o número de funcionalidades com erros detectados na fase de testes sobre o número de funcionalidades a serem entregues.

4.1.4 Aumentar a satisfação do cliente

O objetivo está relacionado com o reflexo do processo de desenvolvimento na qualidade do produto (software), partindo do ponto de vista do cliente. Para o objetivo, foram definidas as seguintes questões e métricas:

a) **Questão Q4.1:** Como reduzir o número de não conformidades detectadas pelo cliente na entrega?

- **Métrica M4.1:** Percentual da razão entre o número de funcionalidades não conformes sobre o número de funcionalidades entregues.

b) **Questão Q4.2:** Como reduzir a quantidade de erros detectados pelo cliente após a entrega?

- **Métrica M4.2:** Percentual da razão entre o número de funcionalidades com erros detectados pelo cliente sobre o número de funcionalidades entregues.

O Quadro 7 apresenta um resumo das questões e métricas em relação aos objetivos definidos. Na coluna “Objetivos” são informados os objetivos de implementação no atual processo de desenvolvimento. Nas colunas “Questão” e “Métrica” são informadas as questões e métricas definidas para cada objetivo.

Quadro 7 - Resumo das questões e métricas por objetivo

Objetivos	Questão	Métrica
Definir e documentar um processo padronizado de desenvolvimento	Q1.1	M1.1
	Q1.2	M1.2
Deixar mais claro o processo de análise de requisitos	Q2.1	M2.1
		M2.2
Aumentar a eficiência da equipe de desenvolvimento	Q3.1	M3.1
	Q3.2	M3.2
	Q3.3	M3.3
Aumentar a satisfação do cliente	Q4.1	M4.1
	Q4.2	M4.2

Fonte: Elaborado pelo autor

As mudanças propostas para o melhoramento do processo atual, foram baseadas nos objetivos, questões e métricas definidos nesta etapa de medição do processo. Após a implantação destas propostas, as métricas serão utilizadas para verificar se os objetivos foram alcançados.

Caso estes objetivos sejam alcançados, a organização passaria do nível de maturidade “Gerenciado”, onde o processo é executado de acordo com políticas organizacionais, para o nível de maturidade “Definido”, onde o processo é padronizado, melhorado continuamente e disseminado por toda a organização.

4.2 ANÁLISE DO PROCESSO ATUAL

A análise do processo atual foi realizada utilizando técnicas sugeridas por Sommerville (2007), como questionários, entrevistas e observação.

O questionário foi utilizado com o objetivo de efetuar o levantamento sobre o perfil da equipe e do processo atual.

As entrevistas são mais informais do que os questionários, possibilitando a abordagem mais ampla dos assuntos tratados. Tanto o questionário quanto as entrevistas, foram utilizados em caráter exploratório e qualitativo.

A observação permite um estudo mais detalhado do processo, através do acompanhamento do mesmo.

4.2.1 Questionário

O questionário foi enviado em meio eletrônico (por e-mail) aos membros da equipe de desenvolvimento estudada, que é composta por dois gerentes, uma Help Desk, e 31 desenvolvedores. A aplicação do questionário foi efetuada com os seguintes objetivos:

- **Primeiro objetivo:** Identificar o perfil e o conhecimento da equipe;
- **Segundo objetivo:** Capturar informações sobre o processo atual e a percepção da equipe em relação ao processo;
- **Terceiro objetivo:** Capturar sugestões de melhoria no processo, por parte da equipe.

O questionário aplicado segue como anexo deste trabalho. A seguir, estão descritas as questões para atender aos objetivos citados:

a) Identificar o perfil e conhecimento da equipe

Foram elaboradas as seguintes questões:

Questão 1) Qual o seu papel na equipe de desenvolvimento?

Questão 2) Possui dificuldades em executar manutenções em softwares ou sistemas nos quais trabalha, por falta de conhecimento das funcionalidades existentes?

b) Capturar informações sobre o processo atual e a percepção da equipe em relação ao processo

Foram elaboradas as seguintes questões:

Questão 3) Considere que uma metodologia de desenvolvimento de software engloba o processo, os métodos e as ferramentas. Você tem clareza dos métodos, técnicas, processos e ferramentas que estão contidos na metodologia utilizada hoje pela equipe de desenvolvimento?

Questão 4) Você considera que a equipe possui um processo de desenvolvimento padronizado e bem definido?

As questões 2, 3 e 4 são uma combinação de questão fechada com questão aberta. Possuem as alternativas “Sim”, “Não”, “Parcialmente” e a questão aberta “Comentários”.

Questão 5) Descreva brevemente o processo de desenvolvimento utilizado pela sua equipe, desde as solicitações dos usuários até a disponibilização do software desenvolvido.

Questão 6) Há documentação para tudo que é desenvolvido pela equipe de desenvolvimento?

Questão 7) O processo atual de trabalho gera muita documentação?

Questão 8) A equipe tem autonomia e poder de decisão sobre as atividades que exerce?

Questão 9) A equipe possui contato diário ou semanal com o cliente?

Questão 10) A equipe se adapta facilmente, quando ocorrem mudanças que afetam o processo de desenvolvimento?

Questão 11) A equipe possui ferramentas ou formas de se comunicar diariamente?

As questões 6, 7, 8, 9, 10 e 11 são uma combinação de questão fechada com questão aberta. Possuem as alternativas “Sim”, “Não”, “Parcialmente” e a questão aberta “Comentários”.

c) Capturar sugestões de melhoria no processo, por parte da equipe

Foi elaborada a seguinte questão:

Questão 12) Escreva em forma de itens, suas sugestões de melhorias para o processo de desenvolvimento atual. Considere métodos, ferramentas, tudo que você entenda que possa melhorar nas atividades que você desempenha diariamente.

Após o retorno dos questionários respondidos, os resultados foram analisados e tabulados. As respostas que possuíam alternativas e a opção “Comentários”, foram consolidadas no Quadro 8. Na coluna “Objetivo” estão identificados os objetivos especificados. Na coluna “Questão”, estão contidos os números das questões, e na coluna “Resultado”, os resultados apurados.

Quadro 8 - Respostas por objetivos e questões

Objetivo	Questão	Resultados			
		Sim	Não	Parcialmente	Comentários
1	2	15,4%	38,5%	46,2%	
2	3	46,2%	46,2%	7,7%	
	4	19,2%	30,8%	50%	
	6	3,8%	57,7%	38,5%	
	7	19,2%	53,8%	23,1%	- Geraria, mas pouca coisa é documentada (3,8%)
	8	30,8%	3,8%	57,7%	- A equipe tem autonomia para realizar mudanças de lógica de programas, mas dificilmente opina sobre como melhorar a forma de trabalho ou o fluxo dos processos (3,8%); - O analista sênior tem autonomia para decidir os ajustes nos sistemas os quais é responsável (3,8%).
	9	53,8%	15,4%	23,1%	- Somente quando é necessário obter informações (7,6%)
	10	34,6%	30,8%	34,6%	
	11	92,3%	0%	7,7%	

Fonte: Elaborado pelo autor

Para a questão aberta sobre o papel na equipe de desenvolvimento, houveram as seguintes respostas:

- Help Desk - 1 resposta;
- Gerente de desenvolvimento - 1 resposta;
- Programador/Desenvolvedor - 4 respostas;
- Analista - 2 respostas;
- Analista/Desenvolvedor - 10 respostas.

Para a questão aberta sobre uma breve descrição do atual processo de desenvolvimento, pode-se concluir que, na opinião dos participantes do questionário, o atual processo de desenvolvimento engloba as seguintes etapas:

- **Solicitação:** Demanda enviada pelo usuário;
- **Análise:** Processo de análise;
- **Desenvolvimento:** Etapa onde as demandas são desenvolvidas;
- **Testes:** Etapa onde são efetuados os testes referentes às demandas desenvolvidas;
- **Entrega:** Disponibilização da demanda solicitada.

Para a questão aberta sobre as sugestões de melhorias por parte dos participantes do questionário, resumidamente as respostas foram:

- Melhorar o contato com o usuário;
- Minimizar ou eliminar o contato direto com o usuário, por parte da equipe de desenvolvimento;
- Implantação de metodologias ágeis;
- Utilização de linguagens modernas para o desenvolvimento Web e Mobile;
- Separação de funções (Analista, Desenvolvedor, Testador, Suporte, Treinamentos, Documentação);
- Melhoramento dos testes e da qualidade;
- Compartilhamento de boas práticas de programação;
- Realização de treinamentos para aperfeiçoamento da equipe;
- Realização de treinamentos com os usuários, para um melhor entendimento de como deve ocorrer a entrada das demandas;
- Trabalhar junto ao usuário, a importância de solicitar alterações que gerem ganhos para as empresas do grupo;
- Documentação dos processos padronizados.

4.2.2 Entrevistas

Em decorrência da pandemia do coronavírus, as entrevistas foram realizadas por meio eletrônico (e-mail). Participaram da entrevista analistas e desenvolvedores que integram a equipe de desenvolvimento. Os principais objetivos da entrevista realizada foram:

- **Primeiro objetivo:** Capturar detalhes do processo atual e da funcionamento da equipe de desenvolvimento, que não foram identificados no questionário;
- **Segundo objetivo:** Identificar as ferramentas utilizadas pela equipe bem como os artefatos gerados durante e em cada etapa do processo de desenvolvimento;
- **Terceiro objetivo:** Verificar a forma de controle e medição utilizada pela equipe no processo atual;
- **Quarto objetivo:** Identificar o parecer e a percepção da equipe em relação aos pontos fortes e pontos fracos existentes no processo de desenvolvimento atual.

O Quadro 9 apresenta um resumo das informações obtidas através da entrevista realizada com os analistas e desenvolvedores integrantes da equipe de desenvolvimento. A entrevista bem como perguntas que foram direcionadas aos colaboradores está anexada ao trabalho. Na coluna “Objetivos” consta a informação de cada objetivo definido para a realização da entrevista, enquanto que na coluna “Resultados obtidos”, consta a descrição e um resumo das informações que foram obtidas.

Quadro 9 - Respostas obtidas na entrevista

Objetivo	Resultados obtidos
Primeiro	Foram capturados detalhes do processo atual como formas de contato com o cliente (usuário), forma como entram as demandas, como é definido e aprovado o que será executado, as atividades realizadas durante o processo, a forma como são disponibilizadas as demandas solicitadas, além do acompanhamento e feedback por parte do cliente (usuário)
Segundo	Identificado o uso das ferramentas Genero, Delphi, PHP, JavaScript, Bizagi, Hangouts e Skype. Os artefatos gerados são solicitações (desenvolvimento ou manutenção), documentações, códigos-fontes, diagramas de processos e relatórios.
Terceiro	Há uma medição de incidentes e uma meta em porcentagem a ser alcançada. Esta medição é efetuada através da quantidade de solicitações entregues e a quantidade de incidentes ocorridos. A medição da produtividade ocorre somente através da quantidade de projetos encerrados, o que não dá uma ideia clara. Não há uma medição do índice de retrabalhos. Há uma data de previsão de entrega para determinado projeto, mas não para determinada solicitação. Esta previsão pode ser alterada a qualquer momento.
Quarto	<p>Pontos fortes:</p> <ul style="list-style-type: none"> - Desenvolvimento ágil e o relacionamento próximo com o cliente (usuário); - Alto conhecimento de alguns integrantes; - A equipe de desenvolvimento está próxima fisicamente de diversas empresas do grupo; - Rápida inclusão de um novo membro à equipe e determinação a ajudar este novo membro. <p>Pontos fracos:</p> <ul style="list-style-type: none"> - Por não envolver custos para as empresas que solicitam novas demandas, por vezes o cliente solicita o desenvolvimento de demandas que não geram custo benefício; - O conhecimento fica restrito aos membros da equipe de desenvolvimento, pois é gerada pouca documentação; - A maioria dos desenvolvedores realizam todas as atividades (Contato com o cliente, Análise, Desenvolvimento, Testes, Disponibilização, Treinamento), fazendo com que não haja o foco em uma atividade específica; - Não há um plano de testes automatizados, fazendo com que hajam poucos testes por parte do desenvolvedor, que geralmente testa somente a funcionalidade desenvolvida.

Fonte: Elaborado pelo autor

4.2.3 Observação

A técnica de observação foi utilizada para capturar detalhes do processo atual e das características da equipe, que não foram identificadas no questionário e nas entrevistas. A aplicação da observação foi realizada em demandas recebidas diariamente e em demandas recebidas via GTs (Grupos de Trabalho).

Esse acompanhamento ocorreu durante alguns dias, onde foram realizadas conversas com o analista e com os programadores de um dos sistemas existentes.

Foi realizado o acompanhamento desde a entrada das demandas até a entrega, analisando atividades como o registro dos projetos e solicitações no sistema Service Desk, manutenção nos softwares existentes, comunicação da equipe durante o processo, entrega e feedback dos clientes (usuários).

Algumas considerações relevantes sobre os aspectos observados são:

- **Comunicação:** É um ponto forte da equipe. A distribuição da equipe é efetuada em ilhas de desenvolvimento, fazendo com que os integrantes estejam próximos uns aos outros, o que faz com que os problemas ou dificuldades sejam rapidamente resolvidas;
- **Contato com o cliente:** É um ponto a ser revisto no processo. Algumas vezes a equipe não conseguiu contato com o usuário, outras vezes o usuário entrou em contato direto com integrantes da equipe, fazendo com que os mesmos não estivessem com o foco total no que estavam desenvolvendo. Estes contatos foram realizados por telefone, por e-mail e por Hangouts;
- **Documentação gerada:** Para registrar as atividades e demais informações sobre o processo de desenvolvimento, a equipe utiliza o sistema Service Desk, desenvolvido internamente. As atividades (solicitações) são criadas pelo cliente (usuário), quando se trata de manutenções necessárias, ou pelo próprio desenvolvedor, quando as demandas entram pelos GTs (Grupos de Trabalho). Quando a demanda é criada pelo usuário, a aprovação da mesma é efetuada pela Help Desk, pessoa responsável por filtrar a entrada das demandas e designar para o desenvolvedor indicado. Quando a demanda entra pelos GTs, o analista do sistema é o responsável por aprová-la;
- **Ferramentas utilizadas:** As ferramentas utilizadas pela equipe são: Genero e Delphi como linguagens de programação, PowerBD (programa desenvolvido internamente) para selecionar os dados no banco de dados. Não é gerada a modelagem de diagramas ER (Entidade-Relacionamento);
- **Testes:** Os testes são manuais e são realizados individualmente. Não é gerada uma documentação ou um processo de testes antes da codificação.

Somente é testada a funcionalidade implementada ou que está sendo alterada. Os testes são realizados em bases de desenvolvimento, onde todos os desenvolvedores realizam manutenções;

- **Liberação da demanda efetuada:** O envio das demandas geralmente é efetuado uma vez por semana, através da “carta de programas”. Se necessário, a demanda é enviada antes do envio semanal. Os clientes (usuários) são informados das novas funcionalidades através da “carta de programas” ou através do sistema Service Desk, no momento em que a solicitação é finalizada. Na maior parte das vezes, pelo que foi observado, não há um retorno por parte do usuário dando um feedback sobre o que foi implementado.

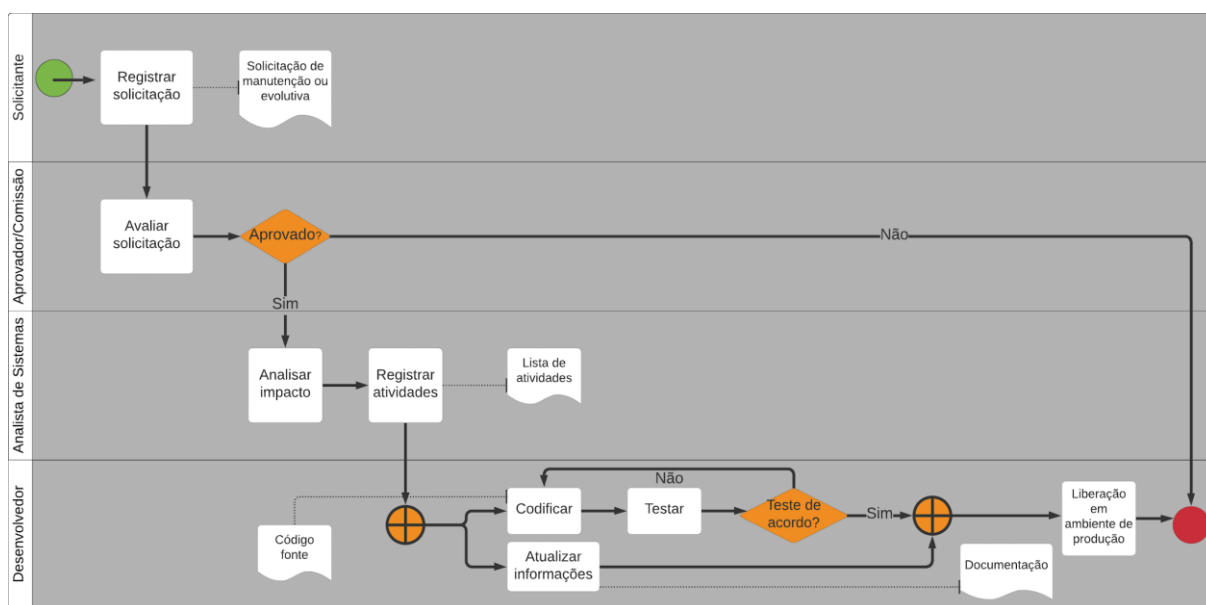
4.2.4 Modelo do processo atual

Através dos resultados obtidos com a aplicação do questionário, das entrevistas e da observação, foram levantadas as informações do processo atual.

A equipe é composta por dois gerentes, uma Help Desk, e 31 desenvolvedores, onde vários deles atuam também como analistas. Além disso, é responsável pelo desenvolvimento e manutenção de diversos sistemas e softwares, que são utilizados por todas as empresas do grupo Tramontina.

Avaliando as atividades do atual processo como um todo, o mesmo pode ser dividido em três fases: Análise, Desenvolvimento e Entrega. A Figura 5, ilustra o modelo do atual processo de desenvolvimento.

Figura 5 - Modelo do atual processo de desenvolvimento



Fonte: Elaborado pelo autor

O processo atual inicia pela fase de Análise, através da atividade “Registrar solicitação”, onde a demanda é registrada pelo cliente (usuário). Esta solicitação é cadastrada em um sistema Service Desk desenvolvido internamente, e pode ser de uma manutenção ou da implantação de uma nova funcionalidade.

Para as solicitações que entram via usuários, a aprovação é efetuada pelos aprovadores de cada sistema e pela Help Desk da equipe de desenvolvimento. Para as solicitações que entram via Grupos de Trabalho (GTs), a aprovação é efetuada pelos integrantes do GT. Essa aprovação é realizada através da atividade “Avaliar solicitação”. Se a solicitação for reprovada, a mesma não é desenvolvida.

Após aprovada a demanda, é efetuada a atividade “Analisar impacto”, que é realizada pelo analista de cada sistema, e registradas as atividades a serem desenvolvidas, através da atividade “Registrar atividades”. No atual processo de desenvolvimento, não são documentados o modelo e o diagrama ER (Entidade-Relacionamento).

Com o devido registro das atividades realizadas pelo analista, a demanda passa para a fase de Desenvolvimento, onde através da atividade “Codificar”, são desenvolvidas pelos programadores as funcionalidades solicitadas. Finalizada esta etapa, são realizados os testes unitários e de sistema, que são realizados pelo

programador através da atividade “Testar”. Se ocorreram erros na etapa de testagem, os artefatos voltam para a atividade “Codificar”. Em paralelo às atividades de codificação e testagem, é realizada a atividade “Atualizar informações”, gerando a documentação do que é desenvolvido.

Por fim, na fase Entrega, são disponibilizadas as funcionalidades desenvolvidas, através da atividade “Liberação em ambiente de produção”. A conferência das funcionalidades disponibilizadas é efetuada por alguns dos usuários, que podem dar um feedback para o analista do sistema.

4.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os resultados dos estágios da medição e análise do processo atual de desenvolvimento.

No estágio de medição do processo, foram definidos os objetivos das melhorias propostas através deste trabalho e, para cada objetivo, foram definidas as questões e métricas, conforme sugere Sommerville (2007).

No estágio de análise do processo atual, foram aplicadas técnicas como questionário, entrevistas e observação. Através dos resultados obtidos a partir da aplicação destas técnicas, foi possível gerar o modelo do atual processo de desenvolvimento.

As informações obtidas na análise do atual processo, serviram de base para a identificação dos aprimoramentos a serem implementados, que estão descritos no próximo capítulo.

5 APRIMORAMENTO DO PROCESSO ATUAL

Sugerido por Sommerville (2007), o ciclo de aprimoramento de processo adotado neste trabalho é composto por estágios de medição, análise e mudança, onde os estágios de medição e análise já foram abordados no capítulo anterior.

O estágio de mudança de processo é composto pelas seguintes fases: identificar aprimoramentos, priorizar aprimoramentos, introduzir mudança de processo, treinar a mudança de processo e ajustar as mudanças de processo. O referencial teórico de cada uma das fases foi apresentado no Capítulo 3, Seção 3.3.

Neste capítulo é apresentada a fase “identificar aprimoramentos”. Não há a necessidade de efetuar a fase “priorizar aprimoramentos” através deste trabalho, pois a proposta é que todos os aprimoramentos sejam implementados ao mesmo tempo. As demais fases serão apresentadas na segunda parte deste trabalho, quando os objetivos serão introduzidos na organização onde o estudo está sendo realizado.

O capítulo é composto por três seções, onde na Seção 5.1 são apresentados os aprimoramentos identificados, partindo dos objetivos elaborados no capítulo anterior. Estes aprimoramentos irão guiar a proposta do novo modelo de processo proposto.

Na Seção 5.2 é apresentada a proposta de melhoria do processo, baseado nos objetivos e aprimoramentos identificados.

Ao final do capítulo, são apresentadas as considerações finais.

5.1 IDENTIFICAR OS APRIMORAMENTOS DO PROCESSO

A primeira fase do estágio de mudança estabelece a identificação dos aprimoramentos, que consiste em utilizar os dados da análise do processo para a definição dos aprimoramentos necessários para alcançar os objetivos definidos.

Nesta fase foram identificados os seguintes aprimoramentos:

- **Definir e padronizar o processo:** A padronização de etapas, atividades, artefatos gerados, ferramentas utilizadas, papéis e responsabilidades, trará mais clareza e fará com que a equipe tenha diretrizes do que e de como fazer. Formalizar um processo padronizado de desenvolvimento facilita a ambientação de novos membros e a velocidade da equipe tende a aumentar. Este aprimoramento vai de encontro ao primeiro objetivo definido pela organização, através da abordagem GQM;
- **Melhorar a etapa de análise dos requisitos:** Este aprimoramento visa diminuir o risco de não conformidades ocorridas na análise dos requisitos referentes às demandas enviadas pelo cliente. A proposta é que seja gerado o artefato “Histórias”, que não é gerado atualmente. Este aprimoramento vai de encontro ao segundo objetivo definido pela organização;
- **Criar uma forma de identificar retrabalhos:** Este aprimoramento visa a identificação das funcionalidades entregues em não conformidade. Com a implementação deste aprimoramento, a equipe terá mais clareza em termos da quantidade e das horas gastas com retrabalhos. Este aprimoramento vai de encontro ao terceiro objetivo definido pela organização;
- **Implementar testes por pares:** Este aprimoramento visa melhorar a etapa dos testes efetuados pela equipe de desenvolvimento, através da testagem sendo efetuada por dois membros da equipe, ou seja, uma testagem dupla das funcionalidades desenvolvidas. Este aprimoramento vai de encontro ao último objetivo definido pela organização.

Para a definição dos objetivos do aprimoramento foram analisados os principais pontos críticos do atual processo de desenvolvimento, que são a falta de um processo de desenvolvimento definido, padronizado e documentado, a alta quantidade de retrabalhos na análise de requisitos e no desenvolvimento de funcionalidades, apesar de não existir um índice de medição atualmente, e o alto índice de erros em funcionalidades entregues.

Fernandes e Teixeira (2004, p. 116) citam que um dos atributos requeridos por uma fábrica de software é a definição e padronização dos processos. Estes atributos foram estudados na seção de atributos de uma fábrica de software, através

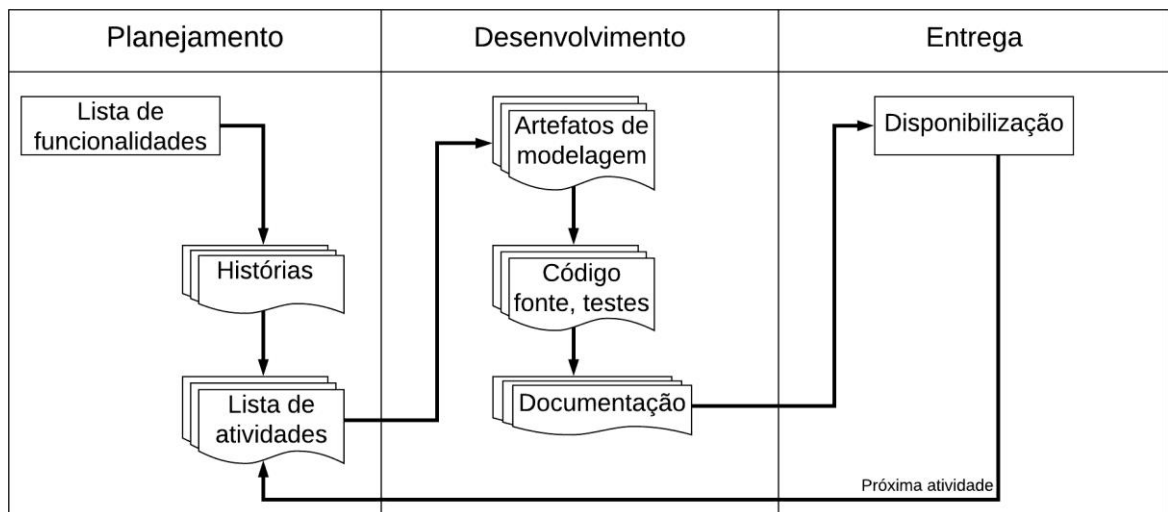
do Capítulo 2. Sobre reduzir o índice de erros na entrega das funcionalidades, essa é uma das formas de reduzir a quantidade de retrabalhos.

Segundo Sommerville (2007), como resultado da fase de identificação de aprimoramentos, deve ser gerado um novo modelo de processo. Esse novo modelo será apresentado na próxima seção.

5.2 O NOVO MODELO DO PROCESSO DE DESENVOLVIMENTO

Partindo dos aprimoramentos identificados na seção anterior, foi montada a proposta do novo modelo do processo de desenvolvimento, o qual será iterativo, e compreenderá as fases de Planejamento, Desenvolvimento e Entrega. Na Figura 6 são apresentadas as fases do processo proposto e os artefatos gerados em cada fase.

Figura 6 - Fases e artefatos do processo de desenvolvimento proposto



Fonte: Elaborado pelo autor

A seguir consta o detalhamento das três fases, além dos papéis envolvidos em cada uma das fases, aprimoramentos propostos, e as atividades e artefatos gerados.

5.2.1 A fase de Planejamento

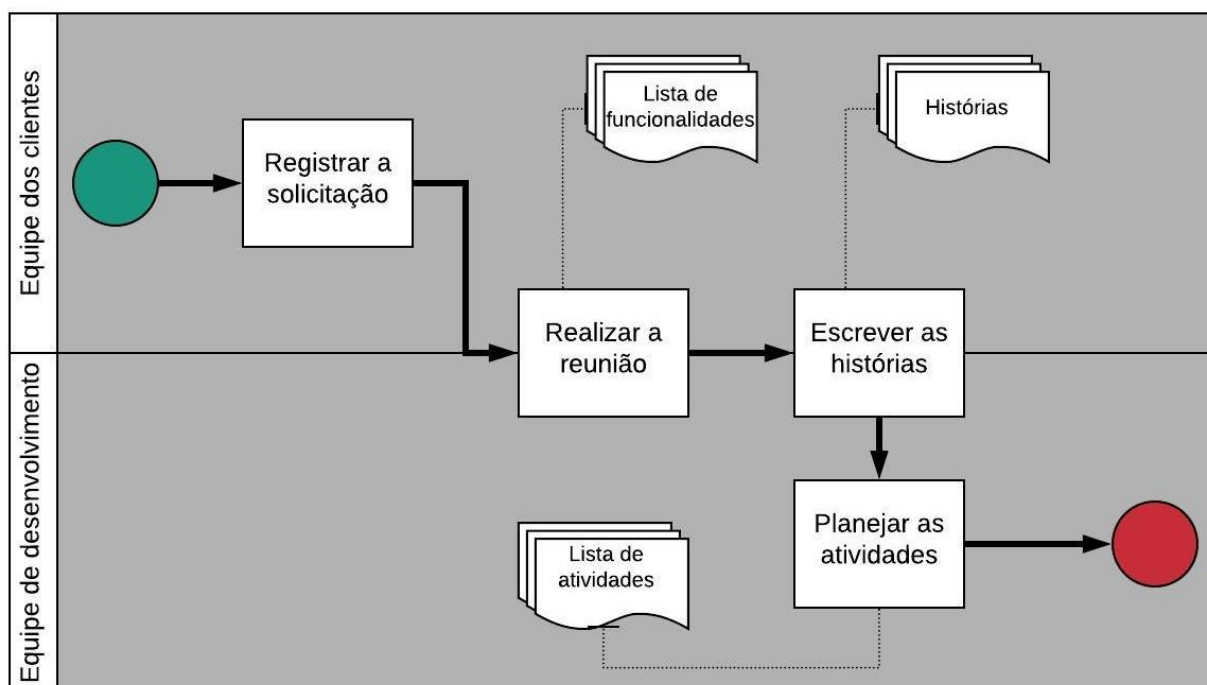
A fase de planejamento tem por objetivo obter junto ao cliente as informações necessárias para o entendimento das reais necessidades e escopo, para que sejam planejadas e estimadas as atividades a serem desenvolvidas.

Os papéis adotados nessa fase são:

- **Equipe de desenvolvimento:** É a equipe técnica de desenvolvimento, composta por analistas, programadores e gerentes;
- **Equipe dos clientes:** É a equipe composta pelos clientes da organização, que corresponde aos usuários que enviam demandas através do sistema Service Desk ou através dos Grupos de Trabalho (GTs).

Na Figura 7 é apresentado o processo executado na fase de planejamento, que recebe como entrada a demanda enviada e gera como saída a lista de atividades.

Figura 7 - Processo proposto para a fase de planejamento



Fonte: Elaborado pelo autor

A fase de planejamento é composta pelas seguintes atividades:

a) Atividade Registrar a solicitação: A atividade corresponde à inclusão da demanda efetuada pelo cliente, que pode ser um único usuário ou um grupo de trabalho.

b) Atividade Realizar a reunião: A atividade corresponde ao contato entre a equipe de desenvolvimento e o cliente, onde o mesmo apresenta suas necessidades e expectativas. Esta atividade gera como artefato a “Lista de funcionalidades”, que consiste em uma lista das funcionalidades que o sistema deve atender. Esta lista é posteriormente priorizada pelo cliente, que deverá escrever uma história para cada funcionalidade. O objetivo da priorização por parte do cliente é a definição das funcionalidades que possuem maior valor de negócio.

c) Atividade Escrever as histórias: A atividade corresponde a geração das histórias escritas pelo cliente junto com o analista. Esta atividade foi inserida no processo de desenvolvimento proposto como forma de reduzir o retrabalho na análise dos requisitos e o índice de incidentes, através da escrita dos testes de sistema pelo analista. O objetivo destas histórias é descrever informações sobre as funcionalidades, como o cliente espera que o sistema ou software se comporte, assim como os requisitos de testes, que serão executados durante a fase de desenvolvimento. Nesta atividade é gerado o artefato “Histórias”, onde constará a explicação de cada funcionalidade, contendo todas as informações que o cliente julgar importante e essencial, além dos critérios de aceitação e dos testes de sistema. Neste artefato, a proposta é que sejam informadas as seguintes informações:

- ID: Identificador numérico e sequencial da funcionalidade;
- Descrição: Descrição da funcionalidade;
- Sistema: Informação do sistema ao qual se refere a funcionalidade;
- Programa: Informação do programa ao qual se refere a funcionalidade;
- Data de criação: Informação da data em que a história foi criada;
- Usuário: Informação do usuário que escreveu a história;

- **História:** Deve ser informada pelo usuário (cliente) a explicação referente à funcionalidade que está sendo detalhada, contendo as informações que o usuário julgar importantes e essenciais;
- **Critérios de aceitação:** O usuário deve descrever os critérios de aceitação para cada história. Estes critérios serão considerados na execução dos testes efetuados na fase de desenvolvimento;
- **Testes de sistema:** Informação a ser preenchida pelo analista de sistema quando executada a atividade de realização dos testes de sistema.

Esta atividade não existe no atual processo de desenvolvimento. A proposta deste trabalho é que a inclusão desta atividade na fase de planejamento melhore o entendimento da equipe de desenvolvimento em relação às demandas solicitadas pelo cliente, diminuindo não conformidades no planejamento e nos sistemas e softwares entregues. Na Figura 8 é apresentado um exemplo do artefato “Histórias”.

Figura 8 - Exemplo do artefato Histórias

Histórias	
ID: 1	Descrição: Cadastrar cursos
Sistema: Desenvolvimento Humano	Programa: DHP090 - Cadastro dos cursos
Data de criação: 13/05/2020	Usuário: Teste
História:	
O código deverá ser numérico e sequencial. Informar descrição, carga horária, tipo e categoria. Permitir realizar alterações no cadastro, exceto o código. Permitir consulta e listagem dos cursos em ordem numérica ou alfabética.	
Critérios de aceitação:	
Cadastrar diversos cursos sem a repetição do código. Geração da consulta e listagem ordenadas pelo código ou pela descrição do curso. Alterar ou excluir um curso. Não deve permitir alterar o código do curso.	
Testes de sistema:	
Data:	Resultado:
13/05/2020	Erro na alteração do curso
13/05/2020	Executado com sucesso

Fonte: Elaborado pelo autor

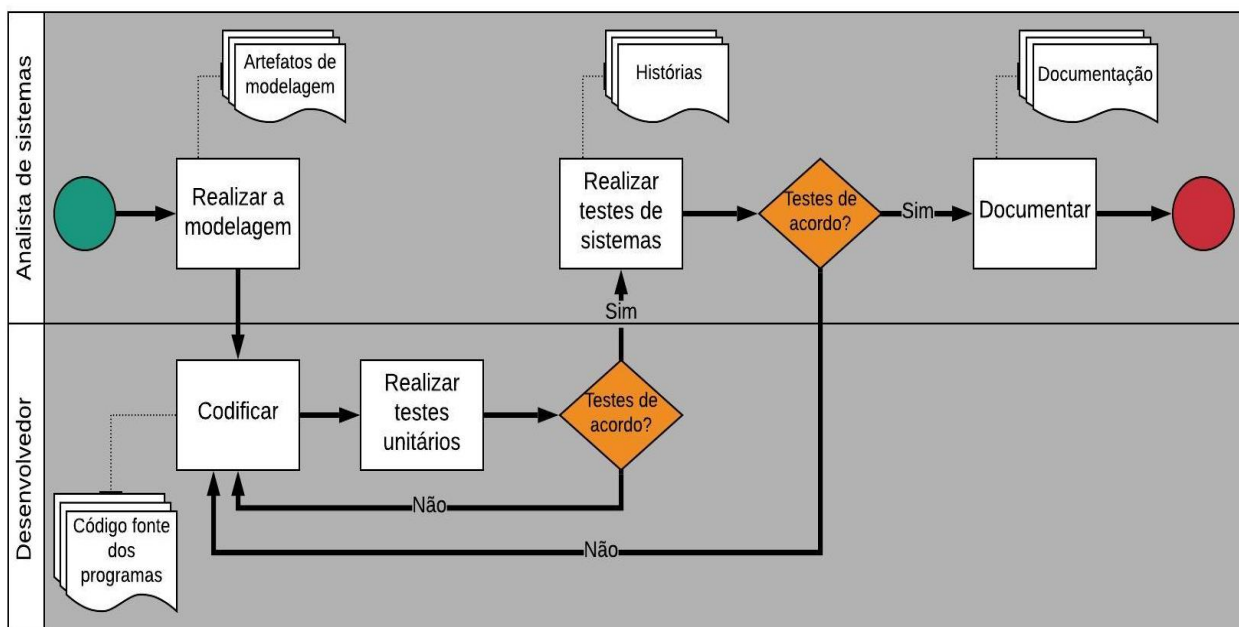
d) Atividade Planejar as atividades: A atividade corresponde a realização de uma reunião entre os integrantes da equipe de desenvolvimento que

irão desenvolver as demandas e os clientes que as solicitaram. O objetivo da reunião é a definição da lista das atividades a serem desenvolvidas para o atendimento das funcionalidades identificadas.

5.2.2 A fase de Desenvolvimento

Nesta fase ocorre o desenvolvimento das demandas solicitadas pelo cliente, através da modelagem, da implementação e dos testes. O processo que é executado na fase de desenvolvimento é cíclico e executado de acordo com a lista de atividades. Quando todas as atividades estiverem concluídas, as funcionalidades passam para a fase da entrega e são disponibilizadas. Na fase de desenvolvimento as atividades geradas são “Realizar a modelagem”, “Codificar”, “Realizar testes unitários”, “Realizar testes de sistema” e “Documentar”. Os papéis envolvidos nesta fase são o analista de sistemas e o desenvolvedor. A Figura 9 apresenta o processo executado na fase de desenvolvimento.

Figura 9 - Processo executado na fase de desenvolvimento



Fonte: Elaborado pelo autor

A fase de desenvolvimento é composta pelas seguintes atividades:

- a) Atividade Realizar modelagem:** Atividade não obrigatória, pois depende da complexidade dos requisitos e do conhecimento do analista para a definição dos artefatos. O objetivo da atividade é gerar a documentação necessária para a equipe de desenvolvimento, compreendendo a análise e o projeto de sistema, e sendo conduzida pelo analista de sistemas. Os artefatos gerados através desta atividade são a estrutura das tabelas no banco de dados e diagramas de entidade e relacionamento, conforme a complexidade do projeto e a real necessidade de geração destes artefatos.
- b) Atividade Codificar:** Atividade que compreende a codificação dos programas, gerando como artefato o código fonte dos mesmos. É o desenvolvedor quem executa esta atividade.
- c) Atividade Realizar testes unitários:** A atividade compreende a execução dos testes unitários, cujo objetivo é a garantia de que o programa está executando as funcionalidades de acordo com o planejado e esperado pelo cliente. A proposta é que estes testes sejam escritos pelo analista, através da geração de um artefato que pode ser um documento. Através da execução da atividade, o desenvolvedor já recebe um feedback imediato sobre o código gerado. Caso sejam identificados erros ou o programa não executar corretamente, o desenvolvedor deve corrigir os erros os desvios de comportamento do programa, submetendo o novo código gerado aos testes unitários. Portanto, esta atividade é realizada de forma cíclica, até o código gerar o resultado esperado.
- d) Atividade Realizar testes de sistema:** A atividade é realizada pelo analista de sistemas e compreende a execução dos testes de aceitação, que devem atender aos critérios de aceitação escritos pelo cliente (usuário) no artefato “Histórias”. Os resultados dos testes devem ser registrados no campo “Testes de sistema” do artefato citado. Assim como a atividade de testes unitários, esta atividade também é realizada de forma cíclica. Caso os resultados não sejam satisfatórios, os programas voltam para a atividade “Codificar” e o desenvolvedor deve rever o código e corrigir as inconsistências verificadas.

e) Atividade Documentar: A atividade também é realizada pelo analista de sistemas e compreende a geração da documentação referente às funcionalidades desenvolvidas. Artefatos gerados através desta atividade são a documentação da versão atual do sistema ou software no ambiente de produção e possíveis documentações sobre problemas enfrentados e soluções adotadas.

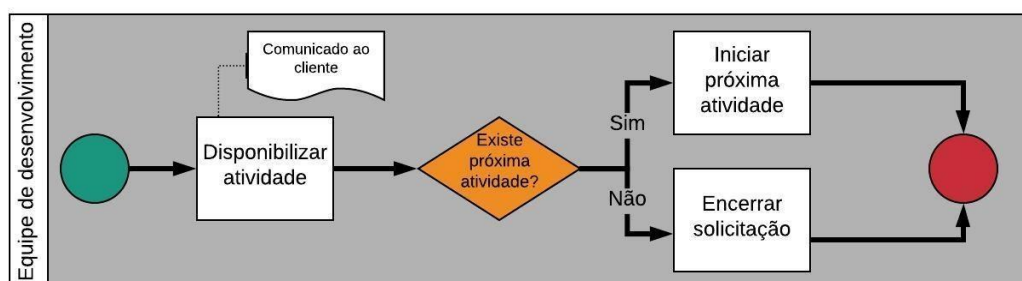
Após a atividade de documentação, o processo para a fase Entrega, que é a última fase do processo. A mesma é apresentada a seguir.

5.2.3 A fase de Entrega

A fase de entrega tem como objetivo disponibilizar as funcionalidades solicitadas pelo cliente. Na mesma ocorre a disponibilização e posteriormente o feedback do cliente. Caso as funcionalidades desenvolvidas não estejam de acordo com o esperado, o cliente cria uma nova demanda para que os ajustes sejam efetuados, e o processo retorna à fase de planejamento. O papel envolvido nesta fase é apenas a equipe de desenvolvimento, formada pelos analistas e desenvolvedores.

Na fase de entrega são executadas as atividades “Disponibilizar atividade”, “Iniciar próxima atividade” e “Encerrar solicitação”. A Figura 10 apresenta o processo executado na fase de entrega.

Figura 10 - Processo executado na fase de entrega



Fonte: Elaborado pelo autor

A fase de entrega é composta pelas seguintes atividades:

- a) Atividade Disponibilizar atividade:** Atividade que corresponde à disponibilização em ambiente de produção da atividade desenvolvida. O artefato gerado nesta atividade é o comunicado ao cliente, que tem como objetivo informar os interessados que as novas funcionalidade estão disponíveis para uso. Atualmente este comunicado ocorre através da “Carta de programas”, que é enviada semanalmente.
- b) Atividade Iniciar próxima atividade:** A atividade corresponde em iniciar a próxima atividade. A mesma remete para a atividade “Lista de atividades” na fase de planejamento, onde a próxima atividade é iniciada.
- c) Atividade Encerrar solicitação:** A atividade corresponde ao fim da solicitação realizada pelo cliente (usuário), que pode ser uma solicitação simples ou um projeto. O encerramento da atividade significa que todas as funcionalidades definidas através da “Lista de funcionalidades” foram desenvolvidas e serão disponibilizadas. A partir deste momento, a equipe envolvida na solicitação poderá assumir uma nova demanda.

5.3 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada a fase “identificar aprimoramentos” do estágio de mudança de processo, seguindo o ciclo de aprimoramento sugerido por Sommerville (2007).

Na fase citada foram detalhados os aprimoramentos identificados, que foram utilizados como base na proposta do novo modelo de processo proposto. Esses aprimoramentos foram apresentados na Seção 5.1, baseados nos objetivos definidos na fase de medição do processo (GQM).

O processo proposto através do desenvolvimento deste trabalho foi apresentado na Seção 5.2, e traz como melhorias a definição e a documentação de um processo padronizado de desenvolvimento, uma nova atividade na análise de requisitos, uma forma de identificar retrabalhos e a implementação de testes por pares. Comparado ao processo anterior, a proposta inclui a atividade de escrita das

histórias, cujo objetivo é trazer mais clareza para a equipe de desenvolvimento em relação às demandas solicitadas pelo cliente.

Definir, padronizar e documentar o processo de desenvolvimento atenderá ao objetivo de padronização do atual processo de desenvolvimento da organização.

Escrever as histórias junto ao cliente atenderá ao objetivo de maior clareza na análise de requisitos, reduzindo possíveis retrabalhos.

Identificar e reduzir os índices de erros e de retrabalho atenderá o objetivo de aumentar a eficiência da equipe de desenvolvimento.

As próximas etapas realizadas neste trabalho, baseadas no ciclo de aprimoramento sugerido por Sommerville foram: introduzir mudança de processo, treinar a mudança de processo e ajustar as mudanças de processo.

6 ESTUDO DE CASO

A metodologia de desenvolvimento proposta através deste trabalho pode ser avaliada com a aplicação de um estudo de caso ou projeto piloto.

Este capítulo apresenta o estudo de caso realizado e está dividido em 4 seções. A Seção 6.1 contempla as características deste estudo de caso, a Seção 6.2 apresenta a aplicação da metodologia proposta e os resultados obtidos, e a Seção 6.3 as considerações finais deste capítulo.

No estudo de caso foram aplicadas as fases “introduzir mudança de processo”, “treinar a equipe” e “ajustar mudanças de processo”, conforme o estágio de mudança e o ciclo de aprimoramento sugerido por Sommerville (2007). Os resultados foram avaliados conforme as métricas definidas na fase de medição do processo, através da abordagem GQM (Objetivo-Questão-Métrica).

Na fase “treinar a equipe” foi realizada uma reunião os integrantes da equipe de desenvolvimento que participam do estudo de caso. Entre estes colaboradores estão um analista e três programadores. Esta equipe reduzida de desenvolvedores utilizou a metodologia proposta em projetos a serem implementados, no período entre agosto e novembro de 2020. Na reunião foi realizado um breve treinamento sobre o processo padronizado a ser seguido na aplicação do estudo de caso, o mesmo segue em anexo neste trabalho.

Na fase “introduzir mudança de processo” o estudo de caso foi aplicado e o processo padronizado sugerido foi posto em prática.

Já na fase “ajustar mudanças de processo” foi realizado o acompanhamento das mudanças propostas. Segundo Sommerville (2007), esta fase deve durar por alguns meses para que ocorram as modificações necessárias e a equipe esteja satisfeita com este novo processo.

6.1 DESCRIÇÃO DO ESTUDO DE CASO

A aplicação do estudo de caso ocorreu no Departamento de Informática da empresa Tramontina Central de Administração, com a participação de uma equipe reduzida de colaboradores, formada por um analista e três desenvolvedores.

Para o estudo foram selecionados 2 projetos de novas demandas e 1 projeto de manutenção evolutiva.

6.1.1 Projeto Benefícios oferecidos pelo Espaço Bem-Estar

A Tramontina, visando o bem-estar dos seus colaboradores, disponibiliza para seus funcionários, dependentes de funcionários e aposentados, os benefícios Zumba, Pilates, Yoga e Funcional.

O objetivo do projeto foi criar um programa para o gerenciamento destes benefícios, com o cadastro das pessoas que os utilizam, cadastro da lista de espera para usufruir dos benefícios, listagem do controle de frequência e geração automática dos boletos, pois quem usufrui paga uma parte do custo. O projeto foi elegido para o estudo de caso porque era uma das prioridades a serem desenvolvidas, além de ser uma nova demanda com alta complexidade, devido aos tipos de pessoas presentes nos benefícios. Além disso, a data de previsão de execução do projeto ficou dentro do período de aplicação deste estudo.

Os usuários que utilizam o programa são os administradores de Desenvolvimento Humano (DH) e os administradores de Recursos Humanos (RH). Como os funcionários aposentados não possuem vínculo com nenhuma das empresas Tramontina, quem gerencia os benefícios usufruídos pelos mesmos são os administradores de DH, já os benefícios usufruídos pelos funcionários e seus dependentes são gerenciados pelos administradores de RH. Este projeto já foi finalizado e disponibilizado.

6.1.2 Projeto Registrar os treinamentos de funcionários temporários

Algumas empresas do grupo Tramontina atualmente contratam funcionários temporários, porém os treinamentos realizados pelos mesmos ainda não estão registrados no sistema de Desenvolvimento Humano, que possui como registro somente os treinamentos realizados pelos funcionários efetivados.

O objetivo deste projeto é efetuar uma manutenção evolutiva no sistema, possibilitando que os treinamentos dos funcionários temporários possam ser registrados. O projeto foi selecionado para o estudo de caso pela grande importância corporativa que o mesmo possui, além de ter alta complexidade e sua previsão de execução estar dentro do período de aplicação deste estudo.

As funcionalidades a serem desenvolvidas através do projeto são possibilitar o cadastro de funcionários temporários nos programas de cadastro das capacitações, além de ajustar os programas que geram as pendências de treinamentos e todos os programas de consulta e listagem das capacitações realizadas pelas empresas da organização.

Os usuários que utilizam os programas são administradores de DH, administradores de RH e administradores de capacitações, que podem ser gestores, pessoas responsáveis por gerenciar as capacitações, integrantes do sistema integrado de gestão, etc. Este projeto ainda está em andamento, porém as atividades, tempo utilizado para o desenvolvimento e os resultados parciais referentes ao projeto serão apresentados nas próximas seções.

6.1.3 Projeto Cadastro dos candidatos via formulário

Antes da pandemia do coronavírus, os candidatos à vagas de emprego nas empresas do grupo Tramontina ligavam para a Central de Currículos da organização e agendavam uma entrevista presencial. Geralmente eram realizadas mais de 200 entrevistas por mês onde, no momento da entrevista, era realizado o cadastro e registrada a avaliação do candidato. Com a pandemia, a impossibilidade das entrevistas presenciais, o pedido de criar uma forma em que cada candidato possa realizar seu cadastro, a necessidade de contratação das empresas e a migração para a utilização das ferramentas Google por parte da organização, surgiu este projeto extra plano com grande urgência por parte do Desenvolvimento Humano.

Sendo assim, os objetivos do projeto foram:

- Criar um cadastro dos candidatos através do formulário Google, onde cada candidato que entrar em contato com a Central de Currículos, recebe o link do formulário para cadastro;
- Criar um Web Service que, a cada inserção de currículo no formulário, realize a inserção deste registro no banco de dados;
- Criar um programa de pré-cadastro dos candidatos, onde os dados inseridos pelo Web Service são processados para o cadastro oficial através da ação do usuário pois, como não há a possibilidade de inserir alguns tipos de validação no formulário, podem existir problemas de cadastro e estes precisam ser solucionados pelo usuário de forma manual antes da inserção no cadastro oficial.

Este projeto foi elegido para o estudo de caso devido a sua urgência para ser implementado, sua alta complexidade e por ter sido iniciado e finalizado dentro do período de aplicação do estudo. Devido ao pouco tempo para concluir este projeto, não foi realizada a etapa “Histórias” junto ao cliente, porém será realizada uma comparação do projeto com os demais, levando em conta possíveis problemas que ocorreram por realizar ou não a etapa citada.

Os usuários envolvidos neste projeto são os administradores de Desenvolvimento Humano e os colaboradores que atuam na Central de Currículos da organização. Esta Central é corporativa e centralizada, portanto os candidatos podem ser contratados por qualquer empresa do grupo Tramontina.

Os resultados obtidos através da aplicação do estudo de caso sobre tais projetos estão descritos na próxima seção.

6.2 APLICAÇÃO DA METODOLOGIA PROPOSTA

Nesta seção são apresentados os resultados obtidos na aplicação do estudo de caso sobre os projetos descritos na seção anterior. Os resultados serão apresentados de acordo com a execução do processo proposto no Capítulo 5, que compreende as fases Planejamento, Desenvolvimento e Entrega.

6.2.1 Fase de planejamento

A fase de planejamento do processo é composta pelas seguintes atividades:

- a) Registrar a solicitação:** A fase de planejamento se iniciou com a criação dos projetos por parte do Desenvolvimento Humano corporativo. Estes projetos foram cadastrados pela Help Desk no programa gerenciador de projetos, programa desenvolvido internamente pela equipe de desenvolvimento. A partir dos projetos criados, foi agendada pelo analista participante do estudo de caso uma reunião para cada projeto, tendo a participação da equipe cliente (usuários) e a equipe de desenvolvimento.
- b) Realizar a reunião:** Nesta etapa, a equipe cliente apresentou suas expectativas em relação ao que deveria ser desenvolvido e também as funcionalidades solicitadas. Como artefato e para cada projeto, foi criada a “Lista de funcionalidades”, que apresenta as funcionalidades solicitadas pelo cliente e que deverão ser executadas pela equipe de desenvolvimento. Não foi definida uma prioridade para cada funcionalidade, pois todas elas foram desenvolvidas de forma paralela a cada projeto executado pelos desenvolvedores. Como responsável pelas funcionalidades foi definido o analista de sistemas. Os Quadros 10, 11 e 12 apresentam a lista de funcionalidades referentes ao projetos.

Quadro 10 - Listas de funcionalidades do projeto Benefícios oferecidos pelo Espaço Bem-Estar

ID	Descrição	Responsável
01	Cadastro dos benefícios	Analista
02	Cadastro das listas de espera	Analista
03	Atualização automática das listas de espera conforme confirmação ou recusa	Analista
04	Listagem do controle de frequências	Analista
05	Consulta e listagem dos benefícios usufruídos pelo colaborador	Analista
06	Consulta e listagem dos benefícios ativos por empresa	Analista
07	Encerramento e abertura automática dos benefícios ao final de cada mês	Analista
08	Geração automática dos boletos de pagamento	Analista

Fonte: Elaborado pelo autor

Quadro 11 - Lista de funcionalidades do projeto Registrar os treinamentos de funcionários temporários

ID	Descrição	Responsável
01	Cadastro das capacitações para os funcionários temporários	Analista
02	Perfil de cargo para os funcionários temporários	Analista
03	Perfil da habilitação para os funcionários temporários	Analista
04	Geração das pendências das capacitações para os funcionários temporários	Analista

Fonte: Elaborado pelo autor

Quadro 12 - Lista de funcionalidades do projeto Cadastro dos candidatos via formulário

ID	Descrição	Responsável
01	Criar o cadastro dos candidatos através do formulário Google	Analista
02	Criar Web Service para a inserção de currículo do formulário no banco de dados	Analista
03	Criar o programa de pré-cadastro dos candidatos	Analista

Fonte: Elaborado pelo autor

c) Escrever as histórias: Dando continuidade ao processo, a próxima atividade realizada foi “Escrever as histórias”. Com a pandemia e o fechamento da Central de Currículos, o projeto “Cadastro dos candidatos via formulário” foi um projeto extra plano e de grande urgência (iniciado e finalizado em aproximadamente 5 dias), portanto para o mesmo não foram escritas as histórias. Na próxima seção serão apresentados problemas gerados justamente pelo fato da atividade não ter sido realizada. Para os demais projetos e em cada uma das reuniões, a atividade foi executada para cada funcionalidade, onde foi criada junto com a equipe cliente uma história com uma breve descrição. Como o artefato “Histórias” ainda não foi implementado no Sistema Tramontina, as histórias geradas foram registradas pelo analista de sistemas em um documento, que não foi anexado ao trabalho por questões de sigilo de informações.

d) Planejar as atividades: Após a equipe de desenvolvimento analisar as funcionalidades a serem desenvolvidas em cada projeto, foram planejadas todas as atividades a serem executadas. Como artefato nesta etapa, foi

criada a “Lista de atividades”. Os Quadros 13, 14 e 15 apresentam os artefatos gerados para os projetos.

Quadro 13 - Lista de atividades do projeto Benefícios oferecidos pelo Espaço Bem-Estar

ID	Descrição	Responsável
01	Criar e alterar as tabelas relacionadas ao cadastro dos benefícios e listas de espera	Desenvolvedor 3
02	Permitir a inclusão de funcionários não associados	Desenvolvedor 2
03	Criar o vínculo dos cursos com os benefícios do Espaço Bem-Estar	Desenvolvedor 1
04	Criar o programa de cadastro e das listas de espera dos benefícios oferecidos pelo Espaço Bem-Estar	Desenvolvedor 1
05	Criar as consultas e listagens dos benefícios	Desenvolvedor 1
06	Geração automática dos boletos	Desenvolvedor 2

Fonte: Elaborado pelo autor

Quadro 14 - Lista de atividades do projeto Registrar os treinamentos de funcionários temporários

ID	Descrição	Responsável
01	Alterar e replicar as tabelas relacionadas aos funcionários temporários e às capacitações	Desenvolvedor 3
02	Cadastro das capacitações para os funcionários temporários	Desenvolvedor 1
03	Perfil de cargo para os funcionários temporários	Desenvolvedor 2
04	Perfil da habilitação para os funcionários temporários	Desenvolvedor 2
05	Geração das pendências das capacitações para os funcionários temporários	Desenvolvedor 1

Fonte: Elaborado pelo autor

Quadro 15 - Lista de atividades do projeto Cadastro dos candidatos via formulário

ID	Descrição	Responsável
01	Criação e alteração das tabelas relacionadas ao cadastro dos candidatos	Desenvolvedor 3
02	Criar o cadastro dos candidatos através do formulário Google	Desenvolvedor 2
03	Criar Web Service para a inserção de currículo do formulário no banco de dados	Desenvolvedor 2
04	Criar o programa de pré-cadastro dos candidatos	Desenvolvedor 1

Fonte: Elaborado pelo autor

Após a criação das listas as atividades foram cadastradas como solicitações de desenvolvimento no sistema Service Desk, e o processo passou para a fase de desenvolvimento, que é apresentada a seguir.

6.2.2 Fase de desenvolvimento

Nesta fase foi executada a análise, a codificação e os testes. A fase ocorreu atividade por atividade, conforme a lista de atividades definida para cada projeto na fase de planejamento, e é composta pelas seguintes atividades:

a) Realizar a modelagem: Esta etapa é facultativa e depende da análise e da complexidade do que deve ser desenvolvido. Para os 3 projetos e como artefato nesta etapa, foi gerada somente a modelagem das tabelas a serem incluídas ou alteradas no banco de dados, pois a equipe de desenvolvimento tem como prática não gerar modelagens como o diagrama Entidade-Relacionamento por exemplo, visando dar maior agilidade ao desenvolvimento das atividades.

b) Codificar: Nesta etapa foram desenvolvidas as atividades criadas através das funcionalidades solicitadas pelo cliente. As horas de trabalho, bem como o tipo de atividade (Análise, Desenvolvimento) foram registradas no sistema Service Desk. Os Quadros 16, 17 e 18 apresentam as atividades e as horas de trabalho para os projetos.

Quadro 16 - Horas de trabalho do projeto Benefícios oferecidos pelo Espaço Bem-Estar

ID	Descrição	Responsável	Horas
01	Criação e alteração das tabelas relacionadas ao cadastro dos benefícios e listas de espera	Desenvolvedor 3	13:07
02	Permitir a inclusão de funcionários não associados	Desenvolvedor 2	04:25
03	Criar o vínculo dos cursos com os benefícios do Espaço Bem-Estar	Desenvolvedor 1	06:03
04	Criar o programa de cadastro e das listas de espera dos benefícios oferecidos pelo Espaço Bem-Estar	Desenvolvedor 1	284:12
05	Criar consultas e listagens dos benefícios	Desenvolvedor 1	52:41
06	Geração automática dos boletos	Desenvolvedor 2	47:55

Fonte: Elaborado pelo autor

Quadro 17 - Horas de trabalho do projeto Registrar os treinamentos de funcionários temporários

ID	Descrição	Responsável	Horas
01	Alterar e replicar as tabelas relacionadas aos funcionários temporários e às capacitações	Desenvolvedor 3	01:02
02	Cadastro das capacitações para os funcionários temporários	Desenvolvedor 1	99:41
03	Perfil de cargo para os funcionários temporários	Desenvolvedor 2	09:57
04	Perfil da habilitação para os funcionários temporários	Desenvolvedor 2	35:48
05	Geração das pendências das capacitações para os funcionários temporários	Desenvolvedor 1	31:03

Fonte: Elaborado pelo autor

Quadro 18 - Horas de trabalho do projeto Cadastro dos candidatos via formulário

ID	Descrição	Responsável	Horas
01	Criação e alteração das tabelas relacionadas ao cadastro dos candidatos	Desenvolvedor 3	04:07
02	Criar o cadastro dos candidatos através do formulário Google	Desenvolvedor 2	10:06
03	Criar Web Service para a inserção de currículo do formulário no banco de dados	Desenvolvedor 2	35:30
04	Criar o programa de pré-cadastro dos candidatos	Desenvolvedor 1	57:30

Fonte: Elaborado pelo autor

c) Realizar testes unitários: Os testes unitários não foram automatizados (codificados) por opção da equipe de desenvolvimento, que os executou de forma manual em todos os projetos. Quando identificados erros ou desvios de comportamento, os desenvolvedores corrigiram os programas e os testaram novamente. Portanto, os desenvolvedores só seguiram adiante quando o resultado dos testes unitários foi positivo.

d) Realizar testes de sistemas: Os testes de sistemas foram executados para cada um dos projetos e em todas histórias escritas. Os mesmos foram guiados pelos critérios de aceitação registrados em cada história. Quando foi detectado algum erro, foi atualizada a situação do critério para “Pendente” até que o programa fosse corrigido e testado novamente. Quando o teste foi executado com sucesso, a situação do critério foi atualizada para “Pronto”.

e) Atividade Documentar: A documentação necessária para os projetos e funcionalidades desenvolvidos foi criada pelo analista de sistemas, utilizando ferramentas como a Wiki Tramontina e o Google Sites. Após a finalização da atividade, o processo passou para a fase de entrega, que é apresentada a seguir.

6.2.3 Fase de entrega

A fase de entrega do estudo de caso proposto é composta pelas etapas “Disponibilizar atividade”, “Iniciar próxima atividade” e “Encerrar solicitação”, conforme a descrição:

- a) Disponibilizar atividade:** Tanto para os projetos já finalizados quanto para o projeto “Registrar os treinamentos de funcionários temporários”, que ainda está em andamento, cada atividade foi disponibilizada conforme a mesma era finalizada. A cada disponibilização de atividade, uma nova atividade a ser desenvolvida era iniciada.
- b) Iniciar próxima atividade:** Nesta etapa e para cada projeto, assim que finalizada uma atividade e enquanto existiam atividades pendentes, uma nova atividade foi iniciada. Quando não houve mais atividades pendentes, o processo passou para etapa de encerramento, que é apresentada a seguir.
- c) Encerrar solicitação:** Esta etapa foi executada somente para os projetos já finalizados. A mesma foi executada quando todas as atividades de cada projeto foram desenvolvidas e testadas.

6.2.4 Medição do processo

A avaliação do modelo proposto neste trabalho foi realizada através da aplicação do estudo de caso e da coleta das métricas definidas na fase de medição do processo. As mudanças de processo propostas foram orientadas pelos objetivos definidos na fase de medição de processo. Conforme sugerido por Sommerville

(2007), as métricas são colhidas após a implantação das mudanças de processo com o intuito de responder às questões e verificar se os objetivos da customização do processo foram alcançados.

Para o estudo de caso aplicado através deste trabalho foram definidos os seguintes objetivos:

- **Definir e documentar um processo padronizado de desenvolvimento;**
- **Deixar mais claro o processo de análise de requisitos;**
- **Aumentar a eficiência da equipe de desenvolvimento;**
- **Aumentar a satisfação do cliente.**

Seguindo a abordagem GQM, para cada objetivo foram definidas questões e, para cada questão, foram definidas métricas. Estas métricas foram aplicadas durante o estudo de caso e são apresentadas no Quadro 19.

Quadro 19 - Métricas aplicadas durante o estudo de caso

Métricas	
M1.1	Razão entre a quantidade de etapas do processo e a quantidade de etapas do processo que estão documentadas.
M1.2	Quantidade de demandas ou projetos que seguem o padrão, através da auditoria realizada.
M2.1	Quantidade de alterações realizadas por requisito, depois da entrada da demanda e entrega ao cliente.
M2.2	Razão entre a quantidade de requisitos modificados sobre a quantidade de requisitos solicitados.
M3.1	Quantidade de alterações efetuadas em demandas ou projetos entregues em N dias.
M3.2	Percentual da razão entre o número de funcionalidades de retrabalho sobre o número de funcionalidades entregues.
M3.3	Percentual da razão entre o número de funcionalidades com erros detectados na fase de testes sobre o número de funcionalidades a serem entregues.
M4.1	Percentual da razão entre o número de funcionalidades não conformes sobre o número de funcionalidades entregues.
M4.2	Percentual da razão entre o número de funcionalidades com erros detectados pelo cliente sobre o número de funcionalidades entregues.

Fonte: Elaborado pelo autor

Para cada métrica foi aplicado o estudo de caso e avaliados os resultados:

- **Métrica M1.1:** Para a aplicação do estudo de caso foram documentadas todas as etapas do processo proposto, cujo documento está anexado neste trabalho. Para o cálculo da métrica foram considerados o total de etapas do processo proposto, 4, e o total de etapas do processo proposto que foram documentadas, 4. Como todas as etapas foram documentadas, a razão entre os totais é igual a 100, o que corresponde a um resultado de 100%. Quanto mais próximo de 100 estiver este percentual, melhor a eficácia da métrica aplicada.
- **Métrica M1.2:** Todos os projetos sobre os quais o estudo foi aplicado seguiram o documento padronizado proposto. Para o cálculo da métrica foram considerados o total de projetos sobre os quais o estudo foi aplicado, 3, e o total de projetos que seguiu o processo padronizado proposto, 3. A razão destes valores é igual a 100, que corresponde a um percentual de 100%. Assim como a métrica anterior, quanto mais próximo de 100 estiver este percentual, melhor será a eficácia da métrica aplicada. Cabe ressaltar que esta métrica passa a ser ainda mais relevante no momento em que for implementado o processo padronizado de desenvolvimento.
- **Métricas M2.1 e M2.2:** Uma das alterações de requisitos ocorrida durante a aplicação do estudo de caso, foi ocasionada pelo fato de não ter sido executada a atividade de planejamento “Escrever as histórias”, para o projeto “Cadastro dos candidatos via formulário”. Para o cálculo das métricas foram considerados o total de requisitos, 13, e o total de alterações realizadas por requisito que ocorreram após a entrada das demandas dos projetos, 2. Após a entrega ao cliente (usuários) não houveram alterações de requisitos. O percentual obtido pela razão destes totais foi de 15%. Assim como para as métricas anteriores, quanto mais próximo de 0 (zero) estiver este percentual, melhor será a eficácia da métrica aplicada.
- **Métrica M3.1:** Durante a aplicação do estudo, não foram necessárias alterações em demandas ou nos projetos que foram entregues. Para o projeto “Benefícios oferecidos pelo Espaço Bem-Estar” já surgiram algumas demandas, porém as mesmas são de novas funcionalidades a serem

desenvolvidas. Desta forma, a quantidade de alterações efetuadas é de 0 (zero). Quanto mais próximo de zero este número, melhor é a eficácia da métrica aplicada.

- **Métrica M3.2:** Um dos retrabalhos que ocorreram durante a aplicação do estudo de caso, foi ocasionado pelo fato de não ter sido executada a atividade de planejamento “Escrever as histórias”, para o projeto “Cadastro dos candidatos via formulário”. Para o cálculo das métricas foram considerados o total de funcionalidades entregues, 11, e o total de funcionalidades de retrabalho, 2. O percentual obtido pela razão destes totais foi de 18%. Assim como para as métricas anteriores, quanto mais próximo de 0 (zero) estiver este percentual, melhor será a eficácia da métrica aplicada.
- **Métrica M3.3:** Uma das propostas de melhoria do processo foi a aplicação da técnica de testes por pares, que foi realizada pelos desenvolvedores. O resultado da aplicação da técnica foi que nenhum erro foi detectado quando aplicada a atividade “Testes de sistemas”. Sendo assim, a razão percentual para a métrica foi de 0% (zero por cento). Quanto mais perto de 0 (zero) este percentual, melhor a eficácia da métrica aplicada.
- **Métrica M4.1:** Nenhuma não conformidade foi verificada pelas equipes cliente após a disponibilização das funcionalidades finalizadas. Portanto, a razão percentual para a métrica foi de 0% (zero por cento). Quanto mais próximo de 0 (zero) este percentual, melhor a eficácia da métrica aplicada.
- **Métrica M4.2:** Nenhum erro foi detectado pelas equipes cliente após a disponibilização das funcionalidades finalizadas. Portanto, a razão percentual para a métrica foi de 0% (zero por cento). Quanto mais próximo de 0 (zero) este percentual, melhor a eficácia da métrica aplicada.

No Quadro 20 são apresentados os cálculos e resultados de cada métrica. Na coluna “Métrica” estão descritas as métricas, na coluna “Cálculo” os cálculos realizados e a coluna “Resultado” apresenta o resultado obtido para cada métrica.

Quadro 20 - Cálculos e resultados por métrica

Métrica	Cálculo	Resultado
M1.1	$4 / 4 = 100$	100%
M1.2	$3 / 3 = 100$	100%
M2.1 e M2.2	$2 / 13 = 0,15$	15%
M3.1		0%
M3.2	$2 / 11 = 0,18$	18%
M3.3	$0 / 11 = 0$	0%
M4.1	$0 / 11 = 0$	0%
M4.2	$0 / 11 = 0$	0%

Fonte: Elaborado pelo autor

Esses resultados correspondem à primeira medição do processo proposto. Conforme Sommerville (2007), o objetivo do estágio de medição é aprimorar as medidas de acordo com os objetivos da organização.

Analisando os resultados obtidos percebe-se que as métricas M2.1, M2.2 e M3.2 devem ser monitoradas, para que estes percentuais sejam reduzidos. Este monitoramento será realizado por projeto. Ambas as três métricas estão relacionadas às atividades de planejamento e levantamento de requisitos, que confirma a tese inicial que a equipe de desenvolvimento está tendo retrabalhos no processo, mesmo que a mesma ainda não possua indicadores para esta questão. Além disso, o fato de um dos retrabalhos ter sido gerado porque não foi executada a atividade “Escrever as histórias” em um dos projetos, confirma que esta mudança no processo proposto pode ser bastante eficaz. Cabe ressaltar que estas métricas terão maior validade quando aplicadas ao processo real de desenvolvimento.

6.3 CONSIDERAÇÕES FINAIS

Este capítulo descreveu a aplicação do estudo de caso com o objetivo de validar o modelo proposto através deste trabalho. O estudo foi aplicado em uma pequena parcela da equipe de desenvolvimento que apresenta as características definidas no objetivo deste trabalho, com a participação de um analista e três

desenvolvedores. Apesar do estudo ter sido relativamente pequeno, seus objetivos foram atingidos. Foi possível verificar que:

- A metodologia de desenvolvimento proposta se aplica a pequenas equipes de desenvolvimento, formadas por um analista e três desenvolvedores, responsáveis tanto por manutenções evolutivas quanto por executar a implantação de novas demandas;
- A proposta da padronização e documentação do processo de desenvolvimento foi considerada extremamente importante pela equipe, pelo fato de que todos os integrantes seguiram o mesmo processo;
- A proposta da implementação da atividade “Escrever as histórias” no processo foi muito bem vinda por parte da equipe, que inclusive identificou retrabalho oriundo do projeto onde a atividade não foi realizada. Segundo os integrantes, as histórias auxiliam em um melhor entendimento das necessidades do cliente, que também compreende melhor o que é o levantamento de requisitos. Ainda segundo eles, a atividade também auxilia na atividade de testes, pois evita por exemplo que algum critério seja esquecido de testar, pelo fato de que os testes realizados são registrados nas histórias a cada critério testado;
- A melhoria de identificação de retrabalhos proposta também foi muito bem vinda pela equipe de desenvolvimento, que considerou de extrema importância ter tais indicadores implementados no Service Desk em um futuro próximo. Segundo eles, tendo os índices de retrabalho juntamente com seus motivos, a equipe pode trabalhar em planos de ações visando a redução destes indicadores;
- Já a melhoria da implementação de testes por pares proposta não foi muito bem vinda pela equipe, mesmo que no estudo de caso ela tenha sido extremamente eficaz. Segundo os integrantes, mesmo a melhoria sendo de utilização opcional, a mesma faz com que a equipe se torne menos eficiente.

Dados os resultados alcançados, entende-se que mesmo o estudo de caso sendo aplicado em uma pequena quantidade de colaboradores, o mesmo possa sim

ser generalizado para equipes maiores e de diferentes projetos, tendo em vista que o mesmo se mostrou eficaz, apesar da pequena amostragem.

7 CONCLUSÃO

O desenvolvimento deste trabalho propiciou estudar, analisar e implementar conceitos e boas práticas de fábrica de software, no atual processo de desenvolvimento da organização que faz parte deste estudo.

Nos Capítulos 2 e 3, foi apresentado o estudo do referencial teórico sobre o conceito e as boas práticas que norteiam uma fábrica de software. Foi elaborado um comparativo entre fábricas tradicionais e de software, um estudo sobre trabalhos correlatos e características de fábrica de software como a organização, a evolução por fases, os atributos, a maturidade, e os processos e metodologias geralmente utilizados em relação ao desenvolvimento de software, aprimoramento de processos e medição da maturidade. As principais conclusões do estudo sobre o conceito Fábrica de Software foram:

- Eliminar o que não agrega valor;
- Simplificar e aperfeiçoar os processos de forma contínua;
- Reutilizar os componentes desenvolvidos;
- Melhorar o planejamento e o gerenciamento de riscos;
- Diminuir o tempo de execução e os custos;
- Reduzir o índice de erros e retrabalho;
- Aumentar a eficiência da equipe de desenvolvimento e a qualidade dos produtos desenvolvidos;
- Padronizar os processos e difundi-los em toda a organização.

Com base na fundamentação teórica desenvolvida, foram definidos junto aos gerentes do Departamento de Informática da empresa Tramontina Central de Administração os objetivos a serem alcançados, levando em conta os principais pontos críticos do atual processo de desenvolvimento, que são a falta de um processo de desenvolvimento definido, padronizado e documentado, alta quantidade de retrabalhos nas etapas de análise de requisitos e desenvolvimento, e o alto índice de erros nas funcionalidades entregues.

Para cada um dos objetivos a serem alcançados foram elaborados os objetivos, questões e métricas, seguindo a abordagem GQM (Objetivo-Questão-Métrica) sugerida por Sommerville (2007). O autor define o aprimoramento como uma atividade cíclica que passa por três estágios: medição de processo, análise do processo e mudança de processo.

Utilizando a abordagem, no Capítulo 4 foram apresentados os resultados da aplicação dos estágios de medição e análise do processo atual. Esta análise permitiu chegar à um modelo do processo atual e identificar os aprimoramentos a serem implementados.

No Capítulo 5 é apresentado o processo de desenvolvimento proposto, que foi norteado pelos objetivos e aprimoramentos identificados na medição e análise do processo atual. O processo proposto foi baseado em um processo iterativo e é composto por três fases: Planejamento, Desenvolvimento e Entrega, onde para cada fase há o detalhamento das atividades a serem executadas.

Com a implementação do processo proposto, o Departamento de Informática atinge um dos critérios para alcançar o nível de maturidade “Definido”, onde o processo é padronizado, melhorado continuamente e disseminado por toda a organização.

Por fim, com o objetivo de validar a metodologia de desenvolvimento proposta, foi aplicado um estudo de caso com alguns dos integrantes da equipe de desenvolvimento, conforme as características definidas no objetivo deste trabalho. Este estudo de caso permitiu validar a eficiência das melhorias propostas, através da utilização de um processo de desenvolvimento padronizado e documentado. Este processo segue como apêndice neste trabalho.

A melhoria proposta “Escrever as histórias” foi eficaz na redução de retrabalho no levantamento de requisitos, tendo em vista que houve pouco retrabalho, um dos retrabalhos foi gerado por mudança de especificação do cliente e o outro foi gerado exatamente no projeto no qual a atividade não foi executada, sendo que a equipe identificou que, se a mesma tivesse sido executada, tal retrabalho não teria sido gerado. A melhoria, segundo a opinião da equipe através de entrevista, também teve grande importância nos índices zerados de erros na entrega pois, como os testes eram

documentados a cada critério de aceitação, nenhum teste foi esquecido de ser executado.

A identificação do que é retrabalho foi avaliada como ponto positivo, tanto na aplicação do estudo de caso quanto para a equipe de desenvolvimento. Segundo a equipe, tais índices e seus motivos ajudam a equipe a identificar porque houve retrabalho, além de promover ações de diminuição destes indicadores.

O ponto negativo relatado pela equipe, apesar de ter sido comprovado como eficiente pelo estudo de caso pois não houve relato de incidentes, ficou por conta da melhoria “Testes por pares”. A mesma foi executada em todos os projetos, deixando a equipe de desenvolvimento menos eficiente. Segundo a equipe, não era necessário que todos os testes fossem duplamente efetuados. Como a atividade é opcional, a utilização da mesma parte da escolha da equipe que for aplicar o processo de desenvolvimento proposto neste trabalho.

Consolidando as experiências durante a aplicação do estudo de caso, o parecer final da equipe de desenvolvimento foi positivo em relação à metodologia de desenvolvimento proposta, pois notou-se uma melhora no entendimento do cliente e no levantamento de requisitos, além da utilização de um processo padronizado e documentado de desenvolvimento, da identificação do que é retrabalho e da redução de incidentes na fase de entrega.

As principais contribuições deste trabalho foram:

- A aplicação da abordagem GQM (Goal-Question-Metric);
- A definição de uma metodologia genérica de desenvolvimento, baseada no ciclo de aprimoramento sugerido por Sommerville (2007), e que pode ser utilizada por equipes de diferentes tamanhos e em distintos projetos;
- A validação do novo processo de desenvolvimento e das melhorias propostas, através da aplicação de um estudo de caso.

A fim de dar continuidade no estudo realizado através deste trabalho, os seguintes trabalhos futuros são destacados:

- Aplicação do processo de desenvolvimento proposto utilizando mais integrantes da equipe de desenvolvimento e distintos projetos;

- Aplicação do processo de desenvolvimento proposto em outras equipes de desenvolvimento, mesmo que as mesmas não tenham as mesmas características da equipe na qual o estudo de caso foi aplicado;
- Desenvolvimento, aplicação e análise estatística de um questionário mais elaborado sobre a avaliação tanto dos clientes como da equipe de desenvolvimento.

REFERÊNCIAS

BENZECRY, Fernando Salztrager. **Metodologias ágeis para gerenciamento de projetos de inovação e pesquisa e desenvolvimento**. Dissertação (Pós-Graduação para Especialista em Gerenciamento de Projetos) – Fundação Getúlio Vargas, 2017. Disponível em: <https://www15.fgv.br/network/tcchandler.axd?tccid=5498>. Acesso em: 10 jun. 2020.

FERNANDES, Aguinaldo Aragon; TEIXEIRA, Descartes de Souza. **Fábrica de Software: Implantação e Gestão de Operações**. São Paulo: Atlas, 2004.

NASCIMENTO, Gustavo Vaz. **Um modelo de referência para o desenvolvimento ágil**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) – Universidade de São Paulo, 2008. Disponível em: https://www.teses.usp.br/teses/disponiveis/55/55134/tde-07052008-170413/publico/dissertacao_bb_digital.pdf. Acesso em: 02 jun. 2020.

NOMURA, Luzia. **Definição e estabelecimento de processos de fábrica de software em uma organização de TI do setor público**. 2008. Tese (Doutorado em Engenharia) – Universidade de São Paulo, 2008. Disponível em: https://www.teses.usp.br/teses/disponiveis/3/3136/tde-30092008-174446/publico/2008_LuzNom_TeseDoutorado.pdf. Acesso em: 01 jun. 2020.

PRESSMAN, Roger. S.; MAXIM, Bruce. R. **Engenharia de Software: Uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016.

POHL, Klaus; Böckle, Günter; van der Linden, Frank J. **Software Product Line Engineering: Foundations, Principles, and Techniques**. Berlin: Springer, 2005.

ROMANHA, Silas Dias. **Um modelo de Fábrica de Software em instituições de ensino superior**. 2016. Dissertação (Mestrado em Engenharia de Produção) – Universidade Estadual Paulista, 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison Wesley, 2007.

TENÓRIO, Fernando Guilherme; VALLE, Rogerio. **Fábrica de Software: Organizadores**. 1. ed. Rio de Janeiro: FGV, 2013.

UNIVERSIDADE DE CAXIAS DO SUL. SISTEMA DE BIBLIOTECAS. **Guia para elaboração de trabalhos acadêmicos**. 6. ed. Caxias do Sul: [S.n.], 2019.

APÊNDICE A - Questionário aplicado na análise do processo atual

LEVANTAMENTO COM A EQUIPE DE DESENVOLVIMENTO

1) Qual o seu papel na equipe de desenvolvimento?

2) Possui dificuldades em executar manutenções em softwares ou sistemas nos quais trabalha, por falta de conhecimento das funcionalidades existentes?

Sim Não Parcialmente

3) Considere que uma metodologia de desenvolvimento de software engloba o processo, os métodos e as ferramentas. Você tem clareza dos métodos, técnicas, processos e ferramentas que estão contidos na metodologia utilizada hoje pela equipe de desenvolvimento?

Sim Não Parcialmente

Comentários:

4) Você considera que a equipe possui um processo de desenvolvimento padronizado e bem definido?

Sim Não Parcialmente

Comentários:

5) Descreva brevemente o processo de desenvolvimento utilizado pela equipe, desde as solicitações dos usuários até a disponibilização do software desenvolvido.

6) Há documentação para tudo que é desenvolvido pela equipe de desenvolvimento?

Sim Não Parcialmente

Comentários:

7) O processo atual de trabalho gera muita documentação?

Sim Não Parcialmente

Comentários:

8) A equipe tem autonomia e poder de decisão sobre as atividades que exerce?

Sim Não Parcialmente

Comentários:

9) A equipe possui contato diário ou semanal com o cliente (usuário)?

Sim Não Parcialmente

Comentários:

10) A equipe se adapta facilmente, quando ocorrem mudanças que afetam o processo de desenvolvimento?

Sim Não Parcialmente

Comentários:

11) A equipe possui ferramentas ou formas de se comunicar diariamente?

Sim Não Parcialmente

Comentários:

12) Escreva em forma de itens, suas sugestões de melhorias para o processo de desenvolvimento atual. Considere métodos, ferramentas, tudo que você entenda que possa melhorar nas atividades que você desempenha diariamente.

APÊNDICE B - Entrevista aplicadas na análise do processo atual

LEVANTAMENTO DO ATUAL PROCESSO DE DESENVOLVIMENTO

Seção 1 - Entradas - Questões relacionadas à entrada das demandas

- 1) De que forma as demandas chegam para a equipe? Existe um padrão?

- 2) Como é definido o que será executado? Há alguém responsável pela aprovação? Qual o fluxo e as pessoas (papéis) envolvidas?

Seção 2 - Processamento - Questões sobre o processo de desenvolvimento

- 3) Como é o processo de desenvolvimento, considerando o levantamento de requisitos, a análise, a modelagem, a implementação e os testes? Descreva as atividades e por quem são executadas.

- 4) Esse processo é padronizado? Sempre ocorre desta forma?

- 5) Como são administrados os problemas e dúvidas que ocorrem durante o desenvolvimento?

- 6) O usuário acompanha o andamento do desenvolvimento das demandas solicitadas? De que forma?

Seção 3 - Saídas (liberação do software) - Questões sobre as saídas

- 7) Quais são as atividades realizadas para a liberação do sistema ou software desenvolvido?

- 8) Existe a realização de testes por parte dos usuários?

- 9) Os usuários são treinados para a utilização do sistema ou software desenvolvido?

- 10) Como o usuário sabe que as demandas solicitadas estão disponíveis? Como ocorre esta divulgação?

- 11) É criada uma documentação para o usuário?

12) Como ocorre o feedback do usuário em relação à demanda que foi entregue?

Seção 4 - Ferramentas e artefatos - Questões relacionadas às ferramentas utilizadas e aos artefatos gerados pelo atual processo de desenvolvimento

13) Quais são as ferramentas utilizadas no processo de desenvolvimento?

14) Para cada ferramenta, quais atividades e artefatos são gerados?

15) Quais os artefatos gerados durante as fases de levantamento de requisitos, análise, modelagem, programação, testes e implantação?

16) Onde são armazenados estes artefatos? É utilizada alguma ferramenta para tal?

17) É gerada alguma documentação? Qual e que tipo? Onde é armazenada?

18) É utilizada alguma ferramenta para o armazenamento?

19) Existe um controle de acesso à documentação? Quem tem acesso?

Seção 5 - Informações gerais - Questões gerais, relacionadas ao atual processo de desenvolvimento

20) O atual processo de desenvolvimento está documentado?

21) Como ocorre o aprendizado sobre o processo atual, quando entra um novo membro na equipe de desenvolvimento?

22) Quais os pontos fortes do processo atual e da equipe?

23) Quais os pontos fracos do processo atual e da equipe? O que você sugere para minimizar ou acabar com os mesmos?

24) Quais são os problemas que você enfrenta diariamente? O que você faz para resolver os mesmos?

25) Você sente a necessidade de melhorias no atual processo de desenvolvimento?

26) Como são estabelecidos os cronogramas e prazos? Onde ficam registrados? Como são atualizados?

27) Quais são os controles realizados no processo em termos de acompanhamento?

28) Quais as metas ou indicadores? Como são atualizados e acompanhados?

29) Existe um controle de erros, produtividade, retrabalho? Quais e como são avaliados?

APÊNDICE C - Processo padronizado aplicado no estudo de caso

Para os projetos elegidos para o estudo de caso siga as instruções abaixo, referentes às etapas envolvidas no desenvolvimento das funcionalidades solicitadas pelo usuário:

- Fase de planejamento:

Após a entrada da demanda e com base na lista de funcionalidades solicitada pelo cliente (usuário), a equipe de desenvolvimento deve efetuar o planejamento das atividades a serem implementadas. Para gerar a lista de atividades que serão desenvolvidas, a equipe deve realizar uma reunião com o(s) cliente(s) ou usuário(s) envolvido(s), executando assim a atividade “Histórias”. Esta atividade corresponde à geração das histórias escritas junto ao cliente (usuário), onde o objetivo é descrever informações sobre as funcionalidades, como o sistema ou software deve se comportar, informações essenciais, critérios de aceitação e os testes que serão executados durante a fase de desenvolvimento. Abaixo segue um exemplo de como deve ser gerado o artefato “Histórias”:

Histórias	
Descrição: Cadastrar cursos	
Sistema: Desenvolvimento Humano	Programa: DHP090 - Cadastro dos cursos
Data de criação: 13/05/2020	Usuário: Teste
História:	
<p>O código deverá ser numérico e sequencial. Informar descrição, carga horária, tipo e categoria. Permitir realizar alterações no cadastro, exceto o código. Permitir consulta e listagem dos cursos em ordem numérica ou alfabética.</p>	
Crítérios de aceitação:	
<p>Cadastrar diversos cursos sem a repetição do código. Geração da consulta e listagem ordenadas pelo código ou pela descrição do curso. Alterar ou excluir um curso. Não deve permitir alterar o código do curso.</p>	
Testes de sistema:	
Data:	Resultado:

Partindo deste levantamento junto ao cliente, a equipe deve gerar a lista de atividades a serem desenvolvidas.

Observações: Sempre que houver mudança nos requisitos inicialmente escritos junto ao cliente(usuário) na fase de planejamento, deve ser sinalizado o motivo do retrabalho e preenchida a planilha com o tempo gasto para replanejar as atividades a serem desenvolvidas.

- Fase de desenvolvimento:

As instruções contidas nesta fase devem ser aplicadas para cada uma das atividades definidas na lista de atividades, pois este processo de desenvolvimento ocorre de forma cíclica. Sendo assim, quando todas as atividades estiverem concluídas, as funcionalidades passam para a fase da entrega e são disponibilizadas. Inicialmente pode ser gerada a modelagem referente à atividade, porém a mesma não é obrigatória, depende da complexidade dos requisitos e do conhecimento do analista para a definição dos artefatos úteis na modelagem.

Após a etapa de criação da modelagem as atividades devem ser codificadas, gerando assim os códigos fonte, e testadas através da execução de testes unitários, testes de sistema ou testes por pares, que garantem que o programa está executando as funcionalidades de acordo com o planejado e esperado pelo cliente (usuário). Dos tipos de testes citados, os testes por pares são de execução obrigatória nas funcionalidades desenvolvidas, pois é uma das propostas de melhoria de processo. Os testes efetuados devem atender aos critérios de aceitação escritos pelo cliente (usuário) no artefato "Histórias". Os resultados dos testes devem ser registrados no campo "Testes de sistema" do artefato citado.

Após cada atividade ser desenvolvida, deve ser gerada a documentação referente à funcionalidade. Ela servirá para o cliente (usuário) como ajuda de utilização do programa e funcionalidade que será disponibilizada. Os artefatos de documentação que podem ser gerados são a documentação da versão atual do sistema ou software no ambiente de produção, problemas enfrentados e soluções adotadas.

Após a atividade de documentação, as funcionalidades desenvolvidas através de cada atividade devem ser disponibilizadas.

Observações: Caso ocorram casos onde houve alteração de requisitos por parte do cliente (usuário), ou foram identificados problemas de execução nas funcionalidades disponibilizadas, deve ser sinalizado na planilha o tempo gasto com retrabalho, informando a data, a funcionalidade e o problema ocorrido.

- Fase de entrega:

Se a equipe está desenvolvendo a última atividade da lista, então a equipe busca uma nova demanda de trabalho. Do contrário, a equipe deve desenvolver a próxima atividade da lista.

APÊNDICE D - Entrevista aplicada após a aplicação do estudo de caso**PERCEPÇÃO DA EQUIPE EM RELAÇÃO AO PROCESSO PROPOSTO**

- 1) Como você avalia a implementação do estudo de caso?
- 2) Como você avalia o processo de desenvolvimento proposto?
- 3) Descreva os pontos positivos e negativos sobre a aplicação do estudo de caso.
- 4) Descreva os pontos positivos e negativos sobre o processo de desenvolvimento proposto.
- 5) Na sua opinião, o processo de desenvolvimento proposto pode ser utilizado por outras equipes e em diferentes projetos?
- 6) O que você melhoraria no processo de desenvolvimento proposto?