

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

ISABELLA MENEZES SANTOS

**DEFINIÇÃO DE UM PROCESSO DE DESENVOLVIMENTO
DE SOFTWARE BASEADO NO SCRUM PARA UMA
EMPRESA DE MÉDIO PORTE CERTIFICADA NÍVEL G DO
MPS. BR**

**ITABAIANA
2011**

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

ISABELLA MENEZES SANTOS

**DEFINIÇÃO DE UM PROCESSO DE DESENVOLVIMENTO
DE SOFTWARE BASEADO NO SCRUM PARA UMA
EMPRESA DE MÉDIO PORTE CERTIFICADA NÍVEL G DO
MPS.BR**

Trabalho de Conclusão de Curso
submetido ao Departamento de
Sistemas de Informação da
Universidade Federal de Sergipe
como requisito parcial para a
obtenção do título de Bacharel em
Sistemas de Informação.

Orientador: Msc. Marcos Barbosa Dósea

**ITABAIANA
2011**

ISABELLA MENEZES SANTOS

**DEFINIÇÃO DE UM PROCESSO DE DESENVOLVIMENTO
DE SOFTWARE BASEADO NO SCRUM PARA UMA
EMPRESA DE MÉDIO PORTE CERTIFICADA NÍVEL G DO
MPS.BR**

Trabalho de Conclusão de Curso submetido ao corpo docente do Departamento de Sistemas de Informação da Universidade Federal de Sergipe (DSI-ITA/UFS) como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Itabaiana, 7 , dezembro e 2011 .

BANCA EXAMINADORA:

**Prof.(o) Marcos Barbosa Dósea, Mestre
Orientador
Universidade Federal de Sergipe**

**Prof.(o) Andrés Ignacio Martínez Menéndez, Mestre
Universidade Federal de Sergipe**

**Prof.(o) Eugênio Rubens Cardoso Braz, Doutor
Universidade Federal de Sergipe**

Dedico

Em memória do meu primo Acácio Santana de Menezes, sua lembrança estará sempre presente em meu coração.

AGRADECIMENTOS

Ao professor Marcos Barbosa Dósea, orientador desse trabalho, pela orientação, esclarecimento e efetivação do mesmo, meus sinceros agradecimentos. Confiar no meu potencial me ajudou a não desistir desse curso, além de me propiciar um caminho de reflexões que me ajudaram a ver o mundo acadêmico com olhos de pesquisadora.

A todos os professores do DSI-ITA, em especial a André Luis Menezes Silva e Methanias Colaço Rodrigues Junior, coordenador (es) da disciplina, que esteve durante todo o ano organizando e esclarecendo dúvidas sobre os trabalhos.

A todos da INFOX Tecnologia da Informação, pelo apoio que me deram durante todo o andamento do trabalho.

Aos amigos de classe, pelo apoio constante e ajuda durante todos esses anos, em especial ao meu amigo Cristiano.

Ao meu pai (*in memoriam*) por iluminar meu caminho em busca de meus objetivos, eu sei que esteve sempre presente e onde estiver está comemorando minha vitória.

A minha mãe, por ser meu exemplo de vida, pelo amor, dedicação e força que tem me dado e apoio sempre nas minhas decisões.

Ao meu noivo, Altran, pelas horas que não pude estar ao seu lado, pelas palavras de apoio e incentivo que sempre me foram dadas durante todo o percurso, o carinho e compreensão que sempre teve comigo.

Agradeço a Deus, meu pai misericordioso, que sempre esteve me guiando para que eu conseguisse chegar até aqui, quem clama por ti não cansa, e o senhor atendeu a todos os meus pedidos, foi uma prova da minha fé que tenho por ti. Obrigado meu pai por tudo.

A todos que não relaciono aqui, amigos e parentes, mas que sempre estiveram ao meu lado e torcem por mim.

Você não sabe
O quanto eu caminhei
Pra chegar até aqui
Percorri milhas e milhas
Antes de dormir
Eu nem cochilei
Os mais belos montes
Escalei
Nas noites escuras
De frio chorei...

(A Estrada – Cidade Negra)

MENEZES, Isabella. **Definição de um Processo de Desenvolvimento de Software baseado no Scrum para uma empresa de médio porte certificada nível G do MPS.BR.** 2011. Trabalho de Conclusão de Curso – Curso de Sistemas de Informação, Departamento de Sistemas de Informação, Universidade Federal de Sergipe, Itabaiana, 2011.

RESUMO

Com o aumento da concorrência de empresas que trabalham desenvolvendo software, a busca pela qualidade dos produtos e serviços vem se tornando cada vez mais um diferencial para a organização que pretende oferecer as melhores soluções, visando à satisfação de seus clientes e o aumento da sua participação no mercado, para competir com empresas já consolidadas no setor da tecnologia da informação. A prática de definição e melhoria de processos de software está sendo adotada por diversas empresas que buscam organizar e controlar os processos dentro do ciclo de vida do software. A melhoria de software consiste na realização de ações que alterem os processos, para que estes satisfaçam, de forma mais eficiente, os objetivos e necessidades das organizações, com a finalidade de reduzir custos, prazos e problemas relacionados a escopo, buscando aumentar sua produtividade, satisfação do cliente, estabelecendo uma linguagem comum entre toda a equipe, construindo uma visão compartilhada e ao mesmo tempo, uma estrutura para priorizar ações e projetos. Este trabalho apresenta um estudo de caso de um processo de desenvolvimento de software baseado no framework Scrum, para ser seguido pela INFOX Tecnologia da Informação Ltda em busca da melhoria dos seus processos, através de uma avaliação do processo atual, no qual foram avaliados os processos de gerência de projetos e gerência de requisitos, realizando a verificação da implementação dos resultados esperados para o nível G do modelo de qualidade MPS.BR.

Palavras-chave: Processos de Desenvolvimento de Software, Metodologias Ágeis, Scrum, Gerenciamento de Projetos, Qualidade, MPS.BR.

ABSTRACT

With increasing competition from companies working to develop software, the search for quality products and services is becoming increasingly a differentiator for the organization that claims to offer the best solutions, to the satisfaction of their customers and increase its participation in market to compete with companies already established in the field of information technology. The practice of defining and improving software processes being adopted by several companies seeking to organize and control processes within the software lifecycle. Improving software consists of performing actions that change the process so that they meet, more effectively, the goals and needs of organizations with the aim of reducing costs, delays and problems related to scope, seeking to increase their productivity, customer satisfaction, establishing a common language throughout the team, building a shared vision and at the same time, a framework for prioritizing actions and projects. This work presents a case study of a software development process based on the Scrum framework, to be followed by INFOX Tecnologia da Informação Ltda in seeking to improve their processes through an assessment of the current process, in which the processes were evaluated project management and requirements management, making verification of the implementation of the results expected for the G model quality MPS.BR.

Key-words: Processes of Software Development, Agile, Scrum, Project Management, Quality, MPS.BR.

Lista de Figuras

Figura 1- Definição do Modelo MPS.BR.....	16
Figura 2- Componentes do Modelo MPS	17
Figura 3 - Níveis de Maturidade do MPS. BR	19
Figura 4 - Demonstração gráfica do processo SCRUM	33
Figura 5 - Fases da Estratégia de Implementação	37
Figura 6 - Método GQM.....	39
Figura 7 - Organograma da INFOX	42
Figura 8 - Organograma da Fábrica de Software	42
Figura 9 - Ciclo de Vida de Desenvolvimento	43
Figura 10 - Fase Iniciação	44
Figura 11 - Fase Planejamento	46
Figura 12 - Fase Elaboração	50
Figura 13 - Fase – Desenvolvimento.....	55
Figura 14 - Fase – Encerramento.....	58
Figura 15 - Fases da Estratégia de Implementação	65
Figura 16 - Fases do Processo Concebido	66
Figura 17 - Fluxo de Atividades da Fase de Pré - Execução.....	67
Figura 18 - Fluxo de Atividades da Fase de Execução.....	70
Figura 19 – Sub-processo Desenvolvimento.....	72
Figura 20 - Fluxo de Atividades da Fase de Encerramento.....	73

Lista de Tabelas

Tabela 1 - Indicativos encontrados para o processo de Gerência de Requisitos	62
Tabela 2 - Indicativos encontrados para o processo de Gerência de Projetos	62
Tabela 3 - Análise do projeto Melhorias Gerais.....	80
Tabela 4 - Análise do módulo Relatórios Gerenciais	80
Tabela 5 - Tempo Médio e Produtividade dos projetos.....	82

SUMÁRIO

1. INTRODUÇÃO.....	13
1.1 O Problema.....	14
1.2 Solução Proposta.....	14
1.3 Contribuições.....	15
1.4 Organização desse Trabalho.....	15
2. FUNDAMENTAÇÃO TEÓRICA.....	16
2.1 MPS.BR.....	16
2.1.1 Estrutura do MPS.BR.....	17
2.1.2 Níveis de Maturidade.....	18
2.1.3 Processos.....	20
2.2 Detalhamento do Nível G.....	22
2.2.1 Gerência de Projetos.....	22
2.2.1.1 Resultados esperados para o Processo Gerência de Projetos.....	22
2.2.2 Gerência de Requisitos.....	28
2.2.2.1 Resultados esperados para o Processo Gerência de Requisitos.....	29
2.3 Metodologias Ágeis.....	30
2.4 SCRUM.....	31
2.4.1 Detalhes do Scrum.....	33
2.4.2 Papéis do Scrum.....	33
2.4.3 Time-Boxes do Scrum.....	34
2.4.4 Processo Scrum.....	36
2.5 Estratégia de Implementação de melhoria de Processo.....	37
2.6 Métricas.....	38
2.6.1 Método GQM.....	39
2.7 Pontos de Função.....	40
3- ESTUDO DE CASO.....	41
3.1 Conhecendo a Empresa.....	41
3.2 Cenário Atual.....	41
3.3 Processo Padrão de Desenvolvimento de Software Infox.....	43

3.3.1 Fases.....	43
3.3.2 Papéis do PPDSI.....	60
3.4 Indicativos Encontrados nos Processos.....	61
3.4.1 Gerência de Requisitos.....	61
3.4.2 Gerência de Projetos.....	62
4- METODOLOGIA DE IMPLEMENTAÇÃO DO PROCESSO.....	65
4.1- Fases da Estratégia de Implementação.....	65
4.2- Fases do Processo Concebido.....	66
4.3- Métricas Utilizadas no Estudo de caso.....	75
5- ESTUDO DE CASO DA APLICAÇÃO DO PPDSI_ÁGIL.....	77
5.1- Sistema Gerenciador de Serviços (SGS).....	77
5.2- Aplicação do Processo Concebido.....	78
5.3- Análise Comparativa entre os Processos.....	78
5.4- Considerações Finais.....	84
6- CONCLUSÃO.....	86
6.1- Principais Contribuições.....	86
6.2- Trabalhos Relacionados.....	87
7- REFERÊNCIAS BIBLIOGRÁFICAS.....	89

1- INTRODUÇÃO

As mudanças que estão ocorrendo nos ambientes de negócios têm motivado as empresas a modificar estruturas organizacionais e processos produtivos, saindo da visão tradicional baseada em áreas funcionais em direção a redes de processos centrados no cliente. A competitividade depende, cada vez mais, do estabelecimento de conexões nestas redes, criando elos essenciais nas cadeias produtivas. Alcançar competitividade pela qualidade, para as empresas de software, implica tanto na melhoria da qualidade dos produtos de software e serviços correlatos, como dos processos de produção e distribuição de software. (SOFTEX, 2009)

Desta forma, assim como para outros setores, qualidade é fator crítico de sucesso para a indústria de software. Para que se tenha um setor de software competitivo, nacional e internacionalmente, é essencial que os empreendedores do setor coloquem a eficiência e a eficácia dos seus processos em foco nas empresas, visando à oferta de produtos de software e serviços correlatos conforme padrões internacionais de qualidade. [SOFTEX, 2009a]

A definição de um processo de desenvolvimento de software é cada vez mais importante para a obtenção dos resultados esperados de um projeto e a qualidade do produto final. Com as possíveis mudanças que podem ocorrer nas empresas relacionadas às pessoas, tecnologias e processo, é importante a definição de um processo que seja gerenciado e melhorado, sendo esse de propriedade da organização.

No Brasil, a SOFTEX foi criada no final de 1996 como uma organização não-governamental cujo objetivo social é apoiar o setor de tecnologia de software no país, com ênfase no mercado externo, visando o desenvolvimento socioeconômico brasileiro, através da inserção do país na economia mundial (SOFTEX, 2009). A SOFTEX é o órgão mantenedor do programa MPS.br – Melhoria de Processo de Software Brasileiro, criado em dezembro de 2003.

O modelo de qualidade MPS.br parte da premissa que todo conjunto de atividades é encarado como um projeto no desenvolvimento de software. Uma de suas metas é definir um modelo de melhoria e avaliação de processo de software, visando preferencialmente às micro, pequenas e médias empresas, de forma a atender as suas necessidades de negócio.

Organizações pequenas acabam por não usar nenhum processo, o que pode trazer efeitos negativos no que diz respeito à qualidade do produto final, além de dificultar a entrega do software nos prazos, custos e funcionalidades previamente definidas.

Para prover o desenvolvimento de software de uma forma mais ágil e conseqüentemente mais acessível ao nível técnico, foram estabelecidos princípios que ficaram conhecidos como o Manifesto ágil [Agile Manifesto, 2001].

Baseado na linha do manifesto ágil foi desenvolvido o Scrum, que aborda a idéia de que o desenvolvimento de software através de métodos ágeis maximiza a entrega do software de maneira eficaz, adaptando-o a mudanças. [ALLIANCE, 2009].

O processo de desenvolvimento da INFOX Tecnologia da Informação Ltda, (INFOX), que é uma empresa de desenvolvimento e integração de sistemas, fundada em 1986, com sede em Aracaju- Sergipe é baseado no modelo cascata com iterações. Entretanto o atual processo não tem contribuído para entrega dos projetos dentro do prazo estabelecido e normalmente com custos e funcionalidades que não atendem completamente ao que foram previamente estabelecidos. Dessa forma torna-se necessária uma revisão no atual processo para identificar problemas que podem estar influenciando no cumprimento dessas metas.

Esse trabalho define um processo baseado no modelo MPS. BR nível G com adoção do framework Scrum, com a aplicação do mesmo em projeto(s) da INFOX, com o objetivo de melhorar as atividades do processo de gerência de projetos e gerência de requisitos no processo de desenvolvimento e atendendo aos requisitos básicos exigidos pelo modelo de qualidade, para assim alcançar melhores níveis de desempenho.

1.1 O Problema

O atual processo da empresa em questão não tem contribuído para entrega dos projetos dentro do prazo estabelecido e normalmente com custos e funcionalidades que não atendem completamente ao que foram previamente estabelecidos.

1.2 Solução Proposta

Definição de um processo baseado no Scrum e alinhado ao MPS.br, com base no processo atual da empresa, contudo atendendo aos requisitos esperados nos processos do nível G do MPS.br.

1.3 Contribuições

As principais contribuições desse trabalho são:

- Implementação do Processo Padrão de Desenvolvimento de Software Infox baseado no Scrum;
- Análise do processo concebido em um projeto de desenvolvimento de software da Infox.
- Análise das necessidades do MPS.br nível G para implantação do Framework Scrum;

1.4 Organização desse Trabalho

O texto referente a este trabalho encontra-se dividido da seguinte forma:

- No Capítulo 2 é feita uma abordagem dos assuntos referentes ao objetivo deste trabalho, para que o leitor identifique e compreenda os principais tópicos abordados e a ligação com o referido tema.
- No Capítulo 3 é mostrado o estudo de caso;
- No Capítulo 4 é mostrada a definição do processo de desenvolvimento de software baseado no Scrum;
- No Capítulo 5 é feita a análise dos resultados obtidos da implementação do processo na empresa;
- No Capítulo 6 são apresentadas todas as considerações e conclusões sobre este trabalho.

2- FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresentamos os assuntos estudados e contemplados neste trabalho, discutimos a composição do modelo de qualidade MPS. BR juntamente com o processo de avaliação e os resultados esperados dos processos referentes ao nível de maturidade G do mesmo, como também aborda Metodologias Ágeis, demonstrando os conceitos e práticas do *Scrum*. A organização do mesmo é dada da seguinte forma: a Seção 2.1 descreve o modelo MPS. BR, a Seção 2.2 detalha o nível G, a Seção 2.3 descreve os princípios das Metodologias Ágeis, a Seção 2.4 detalha o Scrum e na Seção 2.5 são descritas as métricas de software.

2.1 MPS. BR

O MPS. BR é um programa mobilizador, de longo prazo, criado em dezembro de 2003, coordenado pela SOFTEX, que conta com apoio do Ministério da Ciência e Tecnologia (MCT), Financiadora de Estudos e Projetos (FINEP), Serviço de Apoio às Micro e Pequenas Empresas (SEBRAE) e Banco Interamericano de Desenvolvimento (BID), como ilustrado na Figura 1, e tem como objetivo a Melhoria de Processo do Software Brasileiro (MPS.br) (SOFTEX, 2009a, p. 4).



Figura 1- Definição do Modelo MPS.BR

Fonte: Softex, 2009

Esse modelo está dividido em três partes (SOFTEX, 2009):

- MR-MPS – Modelo de referência para melhoria do processo de software (7 níveis de maturidade e suas áreas de processo);
- MA-MPS – Método de avaliação para melhoria do processo de software (Guia de Avaliação 2009), descreve os requisitos para os avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA);
- MN-MPS – Modelo de negócio para melhoria do processo de software (discorre sobre o protocolo para a implementação do MR-MPS pelas Instituições Implementadoras (II), a avaliação seguindo o MA-MPS pelas instituições avaliadoras (IA), organização de grupos de empresas para implementação do MRMPS e avaliação MA-MPS pelas Instituições Organizadoras de Grupos de 17 Empresas (IOGE), certificação de consultores de aquisição e programas anuais de treinamento por meio de cursos e *workshops* MPS.BR).

A seção seguinte aborda a estrutura do MPS.br, detalhando cada componente.

2.1.1 Estrutura do MPS.BR

Os três componentes pertencente ao modelo MPS e seus desdobramentos estão ilustrados na Figura 2.

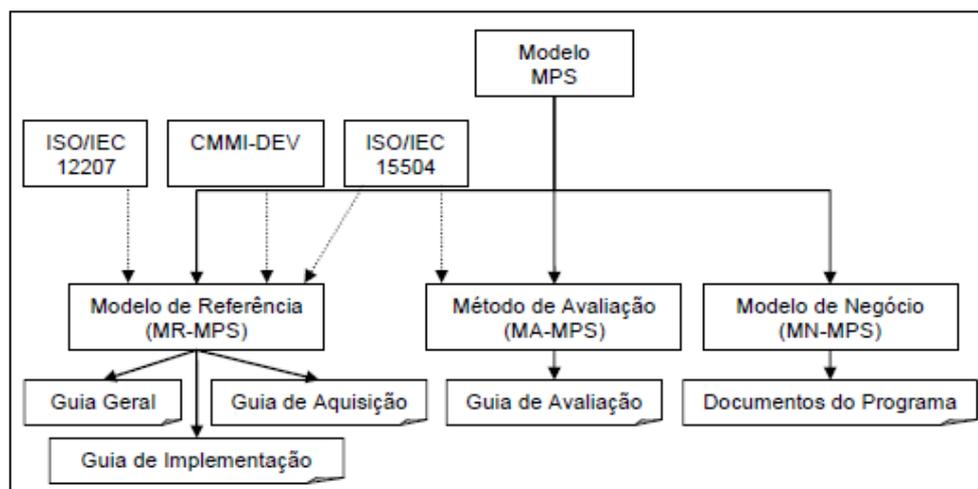


Figura 2- Componentes do Modelo MPS

Fonte: Guia Geral MPS.BR, 2009

O primeiro componente do modelo é o MR-MPS (Modelo de Referência) contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o modelo, nele se encontra o Guia Geral, que contém as definições dos níveis de maturidade, processos e atributos dos processos, o Guia de Aquisição, que é um documento complementar destinado a organizações que desejam adquirir software e serviços correlatos, procurando dar uma orientação quando da aquisição desses produtos e/ou serviços e o Guia de Implementação que é composto de sete partes que descrevem como implementar cada um dos níveis do MR-MPS, detalhando os processos contemplados nos respectivos níveis de maturidade e os resultados esperados com a implementação dos processos.

Já o segundo componente é o MA-MPS (Método de Avaliação), o qual se encontra o Guia de Avaliação, que contém os requisitos para os avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA), com os métodos e formulários.

Dentre os componentes do modelo têm-se o MN-MPS (Modelo de Negócio), o qual descreve as regras de negócio para implementação do MR-MPS pelas II (Instituições Implementadoras), e avaliação seguindo o MA-MPS pelas IA (Instituições Avaliadoras).

De acordo com o modelo existem sete níveis de maturidade, os quais são abordados na próxima seção.

2.1.2 Níveis de Maturidade

Como um dos componentes essenciais do modelo o MR-MPS estão seus níveis de maturidade que são: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado), como demonstrado na figura 3.

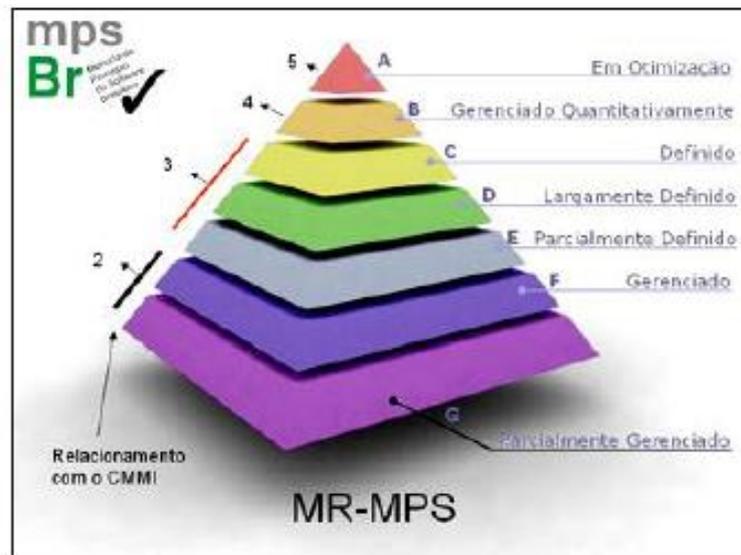


Figura 3 - Níveis de Maturidade do MPS. BR
Fonte: SOFTEX, 2009c

O nível de maturidade é a combinação entre processos e sua capacidade, este corresponde ao grau de melhoria de processo para um pré-determinado conjunto de processos no qual todos os objetivos dentro do conjunto devem ser atendidos. Para cada um destes sete níveis de maturidade é atribuído um perfil de processos que indicam onde a organização deve colocar o grau de melhoria. O progresso e o alcance de um determinado nível de maturidade do MR-MPS se obtêm quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e os resultados esperados dos atributos de processo estabelecidos para aquele nível.

No entanto nota-se que existe uma correlação entre os níveis de maturidade do MPS. BR com o CMMI (Modelo Integrado de Maturidade e Capacidade) que é o modelo de qualidade internacional, onde os níveis G e F correspondem ao nível 2 do CMMI, os níveis E, D e C correspondem ao nível 3, o B ao nível 4 e o A ao nível 5.

A divisão em sete estágios tem o objetivo de se adequar à realidade das empresas brasileiras, mais especialmente micros, pequena e médias empresas. Segundo a SOFTEX (2009a, p. 16), isto possibilita se realizar avaliações com resultados de melhoria de processos em prazos mais curtos.

Dentro de cada nível têm os processos referentes ao mesmo, no qual será abordado na próxima seção, que garante o cumprimento exigido pelo modelo.

2.1.3 Processos

Para cada nível de maturidade existe um perfil de processos com requisitos para a organização concentrar esforços para melhorias (SOFTEX, 2009a). Estes processos estão descritos na seguinte forma:

- Propósito – Descreve o objetivo geral a ser atingido na execução do processo.
- Resultados esperados – Estabelece os resultados a serem obtidos com a efetiva implementação do processo.

Segundo a SOFTEX (2009a, p. 23), alguns processos presentes no MR-MPS podem ser excluídos total ou parcialmente, do escopo de uma avaliação por não serem pertinentes ao negócio da organização que está sendo avaliada. Por exemplo, em organizações do tipo Fábrica de Software é permitida na avaliação a exclusão dos processos referentes à Aquisição.

À medida que a organização avança na escala de níveis do MPS.br, ela tem um maior grau de refinamento na execução de seus processos. Este critério é denominado Capacidade do Processo, e é representado por um conjunto de atributos de processos (AP) e seus respectivos resultados esperados (RAP). Os atributos de processo são requeridos de acordo com o nível de maturidade correspondente em conformidade com seus resultados esperados, e são acumulativos, ou seja, para obter a certificação do nível “E” por exemplo, a organização deve possuir todos os atributos para os níveis “G” e “F”. No nível G do MPS.br são considerados os atributos de processo AP 1.1 – O Processo é executado e AP 2.1 – O processo é gerenciado. Segundo a SOFTEX (2009b, p. 32), em seu MA-MPS, para se considerar um processo satisfeito no nível G o atributo AP 1.1 é necessário estar caracterizado como T (Totalmente implementado) e o atributo AP 2.1 T (Totalmente implementado) ou L (Largamente implementado). Estes níveis de Capacidade do Processo são descritos a seguir (SOFTEX, 2009a, p. 17-18):

○ **AP 1.1 O processo é executado**

- Este atributo é uma medida do quanto o processo atinge o seu propósito.
- Resultado esperado:
 - **RAP 1.** O processo atinge seus resultados definidos.

○ **AP 2.1 O processo é gerenciado**

- Este atributo é uma medida do quanto à execução do processo é gerenciada.
- Resultados esperados:
 - **RAP 2.** Existe uma política organizacional estabelecida e mantida para o processo;
 - **RAP 3.** A execução do processo é planejada;
 - **RAP 4.** (Para o nível G). A execução do processo é monitorada e ajustes são realizados;
 - **RAP 4.** (A partir do nível F). Medidas são planejadas e coletadas para monitoração da execução do processo e ajustes são realizados;
 - **RAP 5.** (Até o nível F) As informações e os recursos necessários para a execução do processo são identificados e disponibilizados;
 - **RAP 5.** (A partir do nível E) Os recursos e informações necessários para executar o processo definido são disponibilizados, alocados e utilizados;
 - **RAP 6.** (Até o nível F) As responsabilidades e a autoridade para executar o processo são definidas, atribuídas e comunicadas;
 - **RAP 6.** (A partir do nível E) Os papéis requeridos, responsabilidades e autoridade para execução do processo definido são atribuídos e comunicados;
 - **RAP 7.** (Até o nível F) As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência;
 - **RAP 7.** (A partir do nível E) As pessoas que executam o processo definido são competentes em termos de formação, treinamento e experiência;
 - **RAP 8.** A comunicação entre as partes interessadas no processo é gerenciada de forma a garantir o seu envolvimento;
 - **RAP 9.** (Até o nível F) Os resultados do processo são revistos com a gerência de alto nível para fornecer visibilidade sobre a sua situação na organização;
 - **RAP 9.** (A partir do nível E) Métodos adequados para monitorar a eficácia e adequação do processo são determinados e os resultados do processo são revistos com a gerência de alto nível para fornecer visibilidade sobre a sua situação na organização;
 - **RAP 10.** (Para o nível G) O processo planejado para o projeto é executado.

- **RAP 10.** (A partir do nível F) A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente e são tratadas as não conformidades.

Na seção seguinte será abordado o nível G do modelo, com o detalhamento dos processos referentes a este nível.

2.2 Detalhamento do Nível G

O nível G é o primeiro nível de maturidade do modelo MPS.BR, neste trabalho abordará somente os processos referentes a este nível, que compreendem a Gerência de Projetos (GPR) e a Gerência de Requisitos (GRE). Neste nível G, a implementação destes processos deve contemplar os atributos de processos AP 1.1 e AP 2.2.

Nas próximas seções, são abordados os propósitos para cada processo do nível G do MPS.BR, segundo o seu guia de implementação.

2.2.1 Gerência de Projetos

O propósito do processo GPR (Gerência de Projetos) é estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito deste processo evolui à medida que a organização cresce em maturidade.

O GPR envolve várias atividades, como: desenvolver um plano geral de controle do projeto; obter o comprometimento e mantê-lo ao longo de toda a execução do projeto; e conhecer o progresso do projeto, de maneira que ações corretivas possam ser tomadas quando a execução do projeto desviar do planejado.

2.2.1.1 Resultados esperados para o Processo Gerência de Projetos

GPR1 – O escopo do trabalho para o projeto é definido

O escopo deve definir todo trabalho necessário, de forma documentada, atendendo todas às necessidades acordadas. Pode ser representado através de uma Estrutura Analítica de Projeto (EAP) (*WorkBreakdownStructure*) (PMI, 2004), Documento de Visão ou outro

documento que defina claramente o escopo (SOFTEX, 2009b, p. 16). A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR1:

Em organizações do tipo Fábrica de Software é importante deixar claro o que é de responsabilidade do contratante do serviço, uma vez que normalmente apenas uma parte do ciclo de vida (codificação) compõe o escopo do projeto que será desenvolvido pela Fábrica de Software; É importante inserir na documentação do escopo, a etapa referente à recepção/compreensão das especificações que, neste caso, constituem uma parte relevante das atividades iniciais de uma Fábrica de Software; É igualmente importante inserir na documentação do escopo quais são as etapas de teste que são desempenhadas pela Fábrica de Software. (SOFTEX, 2009b, p. 16).

GPR2 – As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados

O escopo definido deve ser decomposto em componentes menores, possibilitando seu dimensionamento. Essa decomposição fornece uma base para mensuração de tamanho, esforço, prazo, e responsabilidades. Aqui, somente o Tamanho é estimado, ficando o esforço e o custo para o processo GPR4. O tamanho é a dimensão das funcionalidades sob o ponto de vista do usuário (SOFTEX, 2009b, p. 16). O uso de técnicas apropriadas para dimensionamento do software somente passa a serem exigidas a partir do nível de maturidade E. No nível G esta estimativa pode ser concebida com base na complexidade, número de requisitos, experiência em dados anteriores ou em técnicas utilizadas pela empresa que se mostrem adequadas ao propósito. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR2:

Em organizações do tipo Fábrica de Software, a estimativa precisa de tamanho é essencial para estabelecer, com segurança, os parâmetros do contrato. Neste caso, esta estimativa também é mais factível de ser obtida, uma vez que o produto já está especificado em nível mais detalhado, o que tende a garantir maior precisão na estimativa. Embora técnicas de estimativa mais elaboradas não sejam exigidas no Nível G do MPS.BR, é muito comum que sejam parte da negociação de contratos em organizações do tipo Fábrica de Software. (SOFTEX, 2009b, p.17).

GPR3 – O modelo e as fases do ciclo de vida do projeto são definidos

O modelo descreve a organização das atividades do processo em fases e define como essas fases estão relacionadas (SOFTEX, 2009b, p. 17). Uma organização pode adotar

um ou mais modelos de ciclo de vida para padronização no uso em seus projetos, ou deixar a cargo do(s) responsável(eis) pela escolha dependendo da natureza do projeto. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR3:

As organizações do tipo Fábrica de Software geralmente executam apenas uma parte do ciclo de vida, que corresponde às atividades de implementação (codificação) e testes unitários ou em nível de módulos; É importante detalhar como o ciclo de vida será organizado e como as entregas parciais ocorrerão, se for o caso. Entregas parciais devem ser planejadas junto à organização contratante, de modo a sincronizá-las com o projeto de desenvolvimento como um todo; Uma etapa essencial neste tipo de organização é a de recepção, compreensão e aceitação das especificações recebidas, que deve preceder o início da codificação. Refira-se a GRE 1 para maiores detalhes desta etapa. (SOFTEX, 2009b, p. 18).

GPR4 – (Até o nível F) O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas

Para organizações que estão implementando o nível G, a base de dados históricos normalmente é estabelecida em modelos aplicados ao tamanho, escopo e outros parâmetros do planejamento. Porém é preciso que já se inicie o registro de dados mais precisos a partir dos projetos executados no nível G (SOFTEX, 2009b, p. 18). A SOFTEX, em seu Guia de 51 Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR4:

Para as organizações do tipo Fábrica de Software é mais viável obter estimativas de maior precisão, uma vez que estas se baseiam em especificações mais detalhadas. Devido ao seu escopo reduzido de atuação (etapa de codificação), as bases históricas tendem a refletir mais precisamente o relacionamento entre tamanho e esforço. (SOFTEX, 2009b, p. 19).

GPR5 – O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos

A definição de pontos de controle permite a revisão e atualização do orçamento e cronograma do projeto. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR5:

Para as organizações do tipo Fábrica de Software é importante definir certas pré-condições para o início de algumas de suas atividades, como, por exemplo, o recebimento das especificações vindas da contratante. Em uma organização do tipo Fábrica de Software, muitas das dependências que podem gerar gargalos são provenientes de atividades da contratante. (SOFTEX, 2009b, p. 19).

GPR6 – Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e identificados

O nível G contempla a identificação e acompanhamento dos riscos sem um gerenciamento completo dos mesmos, papel este desempenhado pelo processo Gerência de Riscos (GRI). Segundo a SOFTEX (2009b, p. 19), uma planilha de riscos, contendo dados como identificador, descrição, probabilidade, impacto e prioridades no seu tratamento, pode ser utilizada para identificação dos riscos, monitoração dos riscos identificados e atualização da lista de riscos do projeto à medida que novos riscos forem sendo identificados. É importante demonstrar que esta planilha está sendo monitorada e atualizada. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR6:

Para as organizações do tipo Fábrica de Software, além dos riscos oriundos do próprio projeto, devem ser levados em consideração os riscos advindos da contratante, como, por exemplo, a não entrega das especificações na data planejada ou até mesmo a entrega de especificações em qualidade inferior ao necessário para o início da codificação. (SOFTEX, 2009b, p. 20).

GPR7 – Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo

Este resultado implica na alocação de recursos humanos com base na análise de suas competências (SOFTEX, 2009b, p. 20). É importante manter um plano de competências relacionado aos cargos e funções da organização, bem como uma lista de colaboradores com as respectivas competências, facilitando assim a alocação dos mesmos nos projetos. Este resultado considera ainda a ações de capacitação durante a execução do projeto.

GPR8 – Os recursos e o ambiente de trabalho necessário para executar o projeto são planejados

Este resultado contempla recursos relacionados à infra-estrutura, logística, ações e requisitos não-funcionais para a execução do projeto. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR8:

Em organizações do tipo Fábrica de Software pode ser necessário planejar os recursos específicos relacionados à compatibilidade com o ambiente da contratante, o que inclui: recursos para os testes unitários ou de módulos, disponibilização de infra-estrutura para acesso

remoto a repositórios da contratante, treinamento em ferramentas de uso da contratante etc. (SOFTEX, 2009b, p. 21).

GPR9 – Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança

Este resultado implica na adoção de uma política para gestão da informação referente ao projeto.

GPR10 – Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos

O objetivo deste resultado esperado é garantir que todos os planos que afetam o projeto estejam integrados e que a dependência entre estes planos tenha sido identificada e levada em consideração durante o planejamento, conciliando o trabalho a ser realizado aos recursos e condições existentes (SOFTEX, 2009b, p. 22). Todas as ações e planos do projeto (cronograma, riscos, custos, recursos humanos, etc.) devem estar integrados de tal forma que permitam um acompanhamento preciso do projeto, revisões adaptações com um mínimo de impacto.

GPR11 – A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados

Este resultado contempla o estudo de viabilidade não somente no início do projeto, mas sua reavaliação durante a execução no sentido de garantir ou não sua continuidade. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR11:

Um aspecto importante a ser considerado na análise da viabilidade no caso de uma organização do tipo Fábrica de Software, é a qualidade das especificações recebidas da contratante e a disponibilidade desta para esclarecimento de dúvidas. Este aspecto pode levar à inviabilidade do projeto. (SOFTEX, 2009b, p. 23).

GPR12 – O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido

Este resultado estabelece um nivelamento de expectativas de todos interessados (*stakeholders*) do projeto no sentido de se firmar um compromisso com o mesmo. A

SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR12:

Para as organizações do tipo Fábrica de Software uma atenção especial na obtenção do compromisso com os participantes externos ao projeto é importante pelo alto grau de integração e dependência entre as atividades de codificação e especificação. (SOFTEX, 2009b, p. 23).

GPR13 – O projeto é gerenciado utilizando-se o Plano do Projeto e outros planos que afetam o projeto e os resultados são documentados

Este resultado estabelece o acompanhamento do projeto, comparando o previsto contra o realizado, verificando o alcance de metas em marcos pré-estabelecidos, detectando e corrigindo problemas, e caracterizando uma das atividades mais importantes do gerenciamento.

GPR14 – O envolvimento das partes interessadas no projeto é gerenciada

Este resultado contempla a participação, o compromisso das partes interessadas e a comunicação dentro do projeto. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR14:

Para as organizações do tipo Fábrica de Software é importante o monitoramento das atividades externas, que dependem da contratante como, por exemplo, a entrega das especificações na data acordada.” (SOFTEX, 2009b, p.25). 54

GPR15 – Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento

Este resultado contempla a revisão em marcos, permitindo a verificação de forma mais ampla do andamento do projeto.

GPR16 – Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas

Este resultado contempla o registro e tratamento de problemas provenientes dos processos de monitoramento (GPR13) e revisão em marcos (GPR15). A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR16:

Para as organizações do tipo Fábrica de Software as dependências críticas, em geral, também estão associadas ao relacionamento com a contratante, que é a organização responsável por prover as especificações. (SOFTEX, 2009b, p.25).

GPR17 – Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão

Este resultado estabelece o elo final que se inicia com o registro e tratamento de problemas provenientes dos processos de monitoramento (GPR13), seguido da revisão em marcos (GPR15) e registro dos problemas identificados (GPR16), contemplando ações de correção destes problemas de forma monitorada (acompanhamento). A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GPR17:

Para as organizações do tipo Fábrica de Software os problemas identificados podem envolver as especificações provenientes da contratante, o que pode implicar em replanejamento e até mesmo renegociação de contrato. Nestes casos, é importante identificar quem será o responsável pela ação corretiva: a Fábrica de Software ou a contratante; Esta identificação pode variar conforme as características do projeto ou devido a condições contratuais de atuação junto à contratante. Nestes casos, a definição da responsabilidade pela ação corretiva pode estar definida e registrada, por exemplo, no contrato ou no plano de projeto. (SOFTEX, 2009b, p. 26).

2.2.2 Gerência de Requisitos

O outro processo que compõe o nível G-MPS.BR é o GRE (Gerência de Requisitos), que tem propósito de gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.

O principal objetivo do GRE é controlar a evolução dos requisitos, este processo gerencia todos os requisitos recebidos ou gerados pelo projeto, incluindo requisitos funcionais e não-funcionais, bem como os requisitos impostos ao projeto pela organização. Envolve atribuições como: documentar as mudanças nos requisitos e suas justificativas, bem como manter a rastreabilidade bidirecional entre os requisitos e produtos de trabalho em geral e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.

2.2.2.1 Resultados esperados para o Processo Gerência de Requisitos

GRE1 – Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando critérios objetivos

Este resultado contempla a definição e entendimento dos requisitos junto aos fornecedores do mesmo, estabelecendo um registro de aceitação com marco para controle de mudanças futuras que devem também ser registradas. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GRE1:

Para as organizações do tipo Fábrica de Software as atividades relacionadas a este resultado envolvem o entendimento e a aceitação das especificações enviadas pela organização contratante, utilizando critérios objetivos. As especificações recebidas constituem, neste caso, os requisitos do projeto. (SOFTEX, 2009b, p. 29).

GRE2 – O comprometimento da equipe técnica com os requisitos aprovados é obtido

Este resultado contempla um compromisso da equipe envolvida no atendimento dos requisitos aprovados.

GRE3 – A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida

Este resultado contempla a existência de mecanismos que permitem rastrear a dependência entre requisitos e artefatos produzidos. A SOFTEX, em seu Guia de Implementação, traz o seguinte comentário para organizações do tipo Fábrica de Software em relação ao processo GRE3:

Para as organizações do tipo Fábrica de Software a rastreabilidade bidirecional dos requisitos envolve a rastreabilidade das especificações recebidas em relação aos produtos produzidos pela própria Fábrica de Software (ex.: código, planos de teste unitário etc.). Neste caso, como a Fábrica de Software não tem acesso aos requisitos originais do projeto, não terá como manter a rastreabilidade destes com os demais produtos por ela produzidos. Há casos em que a contratante impõem à Fábrica de Software o seu padrão de gerenciamento da rastreabilidade e, nestes casos, este fato poderá estar registrado no contrato ou no plano de projeto. (SOFTEX, 2009b, p. 30).

GRE4 – Revisões em planos e produtos de trabalho do projeto são realizadas visando a identificar e corrigir inconsistências em relação aos requisitos

Este resultado contempla a validação dos artefatos produzidos no projeto em relação aos respectivos requisitos.

GRE5 – Mudanças nos requisitos são gerenciadas ao longo do projeto

Este resultado contempla o gerenciamento de mudanças em requisitos que contempla a modificação, criação e exclusão de requisitos e o impacto causado no projeto. Aqui o mecanismo de rastreabilidade existente tem papel fundamental.

Na próxima seção será abordado o propósito das metodologias ágeis, para que o processo desenvolvido incorpore práticas e conceitos do mesmo.

2.3 Metodologias Ágeis

A proposta de metodologias ágeis de desenvolvimento começou a surgir na década de 1990 em virtude da insatisfação das equipes de desenvolvimento frente às abordagens clássicas, que de fato são bem mais prescritivas, pois seguem um plano bem definido de execução e possuem alto grau de formalização, com documentação extensa.

No ano de 2001, diante dos problemas gerados nos modelos tradicionais de desenvolvimento de software, tais como: mudanças nas especificações, documentação sem necessidade, falta de atualização da documentação, falha na utilização dos recursos, processo definido, mas não seguido, os entusiastas do assunto se reuniram e formaram o grupo Agile Alliance (AGILE, 2001), onde naquele momento foi formulado um conjunto de princípios chamado Manifesto Ágil (*Manifesto for Agile Software Development*), que nos diz o seguinte:

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através desse trabalho, passamos a valorizar: Indivíduos e interação entre eles mais que processos e ferramentas; Software em funcionamento mais que documentação abrangente; Colaboração com o cliente mais que negociação de contratos; Responder a mudanças mais que seguir um plano. Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda. (AGILE, 2001).

Com base em concordâncias nas práticas e experiências dos envolvidos, foram definidos os princípios que norteiam as abordagens que seguem a linha de desenvolvimento ágil:

1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software com valor;
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas;
3. Entregar software funcionando com frequência, em ciclos de semanas até meses, com preferência aos períodos mais curtos;
4. Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto;
5. Construir projetos com uma equipe motivada. Dando a ela o ambiente e suporte necessário, e confiar que a mesma fará seu trabalho;
6. O método mais eficiente e eficaz de transmitir informações para e entre um time de desenvolvimento, é através de uma conversa cara a cara;
7. Software funcional é a medida primária de progresso;
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter, indefinidamente, passos constantes;
9. Manter contínua atenção a excelência técnica e bom design, aumentando a agilidade;
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito;
11. As melhores arquiteturas, requisitos e designs emergem de times auto-gerenciáveis;
12. Em intervalos regulares, o time avalia em como ficar mais produtivo, ajustando-se e otimizando sua rotina de trabalho. (AGILE, 2001).

Segundo (MARTINS, 2007, p. 316) os Métodos Ágeis são baseados na noção de desenvolvimento e entregas incrementais, de natureza empírica e iterativa, e cada um propõe seu processo específico. Entre os métodos mais conhecidos, temos oXP (eXtremeProgramming), Scrum, e o FDD (FeatureDrivenDevelopment).

No entanto os valores dos métodos ágeis são: comunicação, simplicidade, retorno (feedback), coragem e humildade. Sendo assim, KOSCIANSKI afirma que a adoção destes métodos propõe adaptabilidade, planejamento contínuo, mudanças rápidas, documentação essencial, enfoque nos aspectos humanos do desenvolvimento, gerando assim iterações curtas de desenvolvimento.

A seção a seguir aborda o conceito e práticas do Scrum, para que possam ser utilizados no processo concebido, e o mesmo atinja melhores níveis de desempenho.

2.4 SCRUM

SCRUM é um processo de desenvolvimento ágil, iterativo e incremental de softwares que inicialmente foi proposto como um estilo de gerência de projetos e ganhou atenção pelos bons resultados obtidos em suas equipes. Após essas observações dos resultados, Jeff Sutherland, em 1993, começou a moldar um framework de processo que mais tarde, em 1995, sua documentação foi formalizada por Ken Schwaber.(Schwaber, 2009).

Apesar de ser bem difundido para desenvolvimento de software, o processo ágil também pode ser aplicado a qualquer projeto que priorize o trabalho em equipe e a entrega do produto em um curto espaço de tempo agregando assim um maior valor ao negócio do cliente o mais breve possível.

O termo Scrum é o nome usado para a reunião de jogadores no *Rugby*, quando eles se organizam em círculo para planejar a próxima jogada. É uma forma de mostrar que o projeto deve ser conduzido em pequenos ciclos, mas com uma visão de longo prazo, que é ganhar o jogo. Tem como característica importante o fato de ser adaptativo e empírico, ao contrário dos métodos e técnicas de gerenciamento de projetos que são prescritivas, onde é necessário se seguir uma seqüência pré-definida de passos, pouco sujeita a mudanças (MARTINS, 2007).

Segundo Schwaber (2009), toda implementação de um processo de controle empírico está apoiada em três pilares:

- **Transparência** – A transparência assegura que os aspectos do processo que afetam o resultado devem ser visíveis para aqueles que administram os resultados. Não só estes aspectos devem ser transparentes, mas também o que está sendo visto deve ser conhecido. Isto é, quando alguém inspecionar um processo e comprova que algo está “certo”, isto deve ser equivalente à sua definição de “certo”.

- **Inspeção** – Os vários aspectos do processo devem ser inspecionados com frequência suficiente para que as variações inaceitáveis no processo possam ser detectadas. A frequência das inspeções tem que levar em consideração que todos os processos são alterados pelo ato das mesmas. Um grande problema pode ocorrer quando a frequência necessária das inspeções começarem a atrapalhar o andamento do processo. Felizmente, esta não parece ser uma coisa comum no desenvolvimento de software. Outro fator importante é a habilidade e agilidade de quem inspeciona os resultados do trabalho.

- **Adaptação** – Se o responsável pela inspeção determina que um ou mais aspectos do processo esteja fora dos limites aceitáveis, e que o produto resultante será inaceitável, ele deve ajustar as rotinas a serem executadas. Este ajuste deve ser feito o mais rápido possível para minimizar os desvios.

Desta forma, vale ressaltar que com o uso de framework ágil, os problemas das práticas de desenvolvimento tornam-os transparentes, para que se possam melhorá-los, e faz uso da inspeção e adaptação para obter melhoria contínua do produto e dos processos de desenvolvimento.

2.4.1 Detalhes do Scrum

- **Product Backlog**

O *product backlog* é basicamente uma lista de requisitos, histórias, coisas que o cliente deseja, descritas utilizando a terminologia do cliente e ordenada em uma lista.

- **Sprint Backlog**

O *Sprint Backlog* significa a lista da Sprint, ou seja, é a lista de funcionalidades a serem desenvolvidas nesta Sprint, pode conter o desdobramento em tarefas, é estimada pela equipe, e não muda durante a Sprint, é definido o que será feito e como fazer.

A Figura 4 demonstra o processo Scrum, mostrando sua dinâmica, destacando as etapas de planejamento, seus ciclos de iteração até a entrega das funcionalidades. Nesta figura, fica perceptível como um projeto de software tem suas funcionalidades (*Product Backlog*) identificadas e divididas em partes menores para cada iteração (*Sprints*), com a finalidade gerar um entregável ao final de cada ciclo. As rotinas de inspeção previstas nas práticas de reuniões e retrospectivas permitem a detecção de problemas de maneira mais rápida e auxiliam à tomada de decisões em relação a mudanças no projeto.

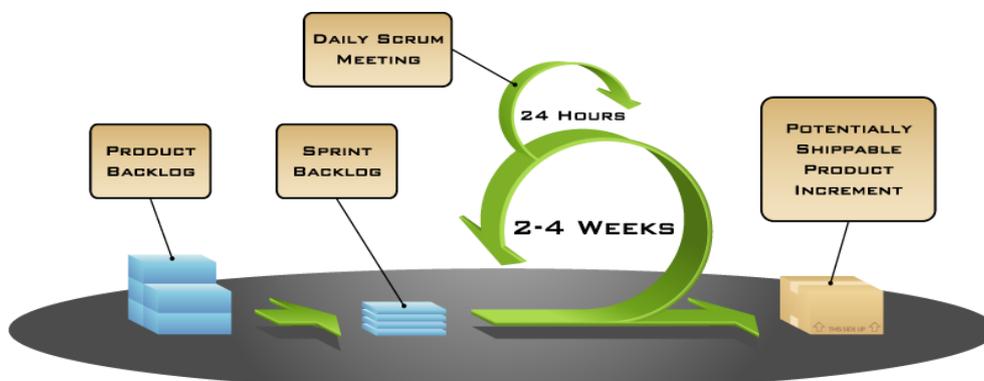


Figura 4 - Demonstração gráfica do processo SCRUM

Fonte: ALLIANCE

Dentro do framework Scrum existem papéis pré-definidos, os quais são detalhados na seção a seguir.

2.4.2 Papéis do Scrum

O Scrum possui apenas três papéis:

- *Scrum Master*

Responsável por garantir que o Time possa entender e aplicar corretamente os valores do Scrum tem como objetivo a melhoria contínua do Time, fazendo com que se produzam produtos sempre com mais qualidade, proporciona ao Time aplicar o autogerenciamento e ajuda na remoção de impedimentos do Time.

- *Product Owner*

É a pessoa responsável pelo gerenciamento do *Product Backlog*, define o valor de cada item (ROI), define quais os itens prioritários conforme o valor de negócio garante que o *backlog* estará visível para todos, e tem a permissão de aceitar ou rejeitar os resultados.

- *Time*

É formado por um grupo entre 5 a 9 pessoas, precisa ser multidisciplinar, transformam o *Product Backlog* em incremento de funcionalidade potencialmente entregável, não há títulos dentro do Time, ou seja, arquiteto, designer, testador, analista, etc, e deve ser auto-organizável.

Para a execução das atividades estabelecidas pelo framework, existem eventos de duração fixa, que será abordado na próxima seção.

2.4.3 Time-Boxes do Scrum

São eventos de duração fixa que possuem determinados objetivos, os times-boxes do Scrum são:

- *Release Planning*

É no *release planning* onde se define a estimativa do número de *sprints* necessários e uma alocação prévia das estórias nos *sprints* gerando uma estimativa inicial de fim do projeto.

Tenta responder as seguintes perguntas:

- Como transformar a visão em um produto de boa qualidade da melhor forma possível?

- Quando teremos uma versão “pronta”?

- Como podemos alcançar ou exceder a satisfação do cliente e o ROI desejados?

- *Sprint Planning*

Este é o *time-box* para o planejamento da iteração, normalmente ocupa-se 5% do tempo do tamanho do *Sprint*, os *time-boxes* são sempre de tamanho fixo.

Para que o *Sprint Planning* possa ocorrer algumas coisas devem estar prontas

- ✓ O *product backlog* deve existir;
- ✓ Deve existir um único *product backlog* e apenas um *product owner*;
- ✓ Todos os itens importantes do *product backlog* devem estar com sua importância estimada;
- ✓ O *product owner* deve entender cada estória e saber o motivo dela estar no *product backlog*.

O resultado da *Sprint Planning* deve ser o seguinte conjunto:

- ✓ Objetivo do *sprint*;
- ✓ Uma lista de membros da equipe e seu grau de comprometimento caso não seja 100%;
- ✓ Um *sprint backlog*;
- ✓ Uma data definida para a apresentação do *sprint*;
- ✓ Data e local definidos para a reunião diária.

- ***Sprint***

O *Sprint* é um *time-box* de 1 a 4 semanas no qual o time do projeto irá produzir uma parte do produto definida pelo cliente. Cada *sprint* deve ter uma meta específica que represente o desejo do cliente em incremento de software para aquele *time-box* específico. Os membros do time da *Sprint* são os responsáveis por estimar os itens que compõem o desejo do cliente e dar a palavra final do que será possível ser desenvolvido naquele *time-box*.

- ***Daily Stand-up***

A reunião diária ou *daily stand-up* é uma reunião de status do projeto, acontece todo dia durante o *sprint*, com duração de 15 minutos e deve acontecer no mesmo local e mesma hora todos os dias. Durante a reunião, cada membro da equipe responde a três perguntas:

- ✓ O que você tem feito desde ontem?
- ✓ O que você está planejando fazer hoje?
- ✓ Você tem algum problema impedindo você de realizar seu objetivo?

- ***Sprint Review***

O objetivo da *Sprint Review* é apresentar o trabalho realizado para os interessados, revendo o trabalho que foi concluído e não concluído. Um trabalho incompleto não pode ser demonstrado.

- ***Sprint Retrospective***

É a oportunidade que o time tem para discutir sobre o que está acontecendo durante a sprint, o *Scrum Master* e os membros do time devem participar da retrospectiva, a participação do *Product Owner* é sempre bem-vinda, mas não obrigatória.

A estrutura da *Sprint Restrospective* é bem simples, esta objetiva identificar o que funcionou bem ou o que precisa ser melhorado. Então, o time visualiza os itens citados, debatem sobre e planejam ações a serem tomadas para a próxima Sprint.

Depois de apresentado os papéis e *time-boxes* do Scrum, abordaremos na próxima seção seu processo, no intuito de estabelecer a seqüência que as atividades existentes no framework são executadas.

2.4.4 Processo Scrum

O processo do *Scrum* envolve três fases: *Pre-Game (Planning)*, *Game (Development)* e *Pos-Game (Closure)* (SCHWABER, 1995, p. 12). A primeira e última envolvem processos com entradas e saídas bem definidos, prescritivos e bem explícitos, que podem compreender mais de uma iteração. O *Game*, fase intermediária, incorpora toda prática e empirismo do *Scrum*. “A maioria dos processos nesta fase são indefinidos, e tratados como uma caixa preta.” (MARTINS, 2007).

No *Pré-Game* são desenvolvidas duas atividades: Planejamento e Arquitetura. No Planejamento é definida uma nova versão para lançar o produto (*release*) com base nas informações iniciais, e organizadas em um *backlog*, junto com uma estimativa inicial de prazo e custo. Na Arquitetura definem-se as bases para construção do produto, ou seja, os itens do *backlog* são projetados com vistas à implementação, porém num nível de abstração maior (MARTINS, 2007). Em resumo, o Planejamento e a Arquitetura constituem as fases de Análise (Conceitualização) e Projeto (*Design*) de forma mais limitada. No Planejamento, os *stakeholders* conseguem ter suas expectativas definidas em relação ao projeto e permite que os mesmo acompanhem sua evolução.

Na fase *Game* se dá toda a ação empírica através das iterações de desenvolvimento que são os *Sprints*. Os *Sprints* têm como objetivo a implementação das funcionalidades de um novo release, respeitando as restrições de prazo, requisitos, qualidade e custos (MARTINS, 2007).

Após a verificação, comandada principalmente pelo *Scrum Master* e o Dono do Produto, de que o produto está em condições de ser distribuído ao usuário final, é declarado que o *release* está encerrado e é iniciada a fase de *Pos-Game*. Aqui são realizados testes adicionais, preparação para distribuição (*Deploy*), documentação, material de treinamento, etc.

Na seção seguinte é abordado o conceito de métricas de software, que pode ser utilizadas quando se desejam medir ou avaliar características pertinentes a um processo.

2.5 Estratégia de Implementação de melhoria de Processo

Para implementação das melhorias de processo necessárias para alcançar diferentes níveis do MPS.BR, utiliza-se a gerência de conhecimento para auxiliar a identificação dos itens importantes que irá determinar as melhorias pertinentes ao processo. (Catunda et al., 2010).

As fases da estratégia utilizada podem ser observadas na Figura 5 e são descritas a seguir.

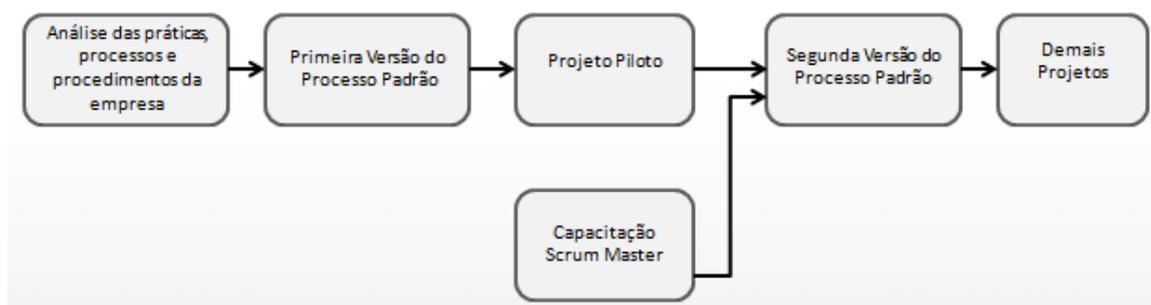


Figura 5 - Fases da Estratégia de Implementação

i. Análise das práticas, processos e procedimentos da empresa: As práticas, os processos e procedimentos já utilizados pela empresa foram analisados, com o objetivo de identificar e entender as necessidades da empresa, permitindo a elaboração de um processo padrão que atendesse a estas necessidades. Neste momento, a empresa colocou a necessidade de manter a agilidade de seus processos adequada ao tamanho dos projetos e ao ciclo de vida utilizado.

ii. Primeira Versão do Processo Padrão: Foi elaborada a primeira versão do processo padrão de desenvolvimento buscando manter as práticas ágeis já utilizadas e atender às necessidades identificadas na fase anterior. O objetivo desta primeira versão é permitir que a empresa

utilize o processo rapidamente em um projeto piloto e aprenda com esta experiência, evoluindo o processo e adequando-o à realidade da empresa.

iii. Projeto Piloto: O processo padrão foi utilizado em um primeiro projeto. Através desta execução do processo, a empresa identificou várias oportunidades de melhoria no processo padrão.

iv. Capacitação Scrum Master: Em paralelo à execução do projeto piloto, a gerência da Fábrica e todos os coordenadores de projetos da empresa fizeram o curso e se tornaram *Certified Scrum Masters*, aumentando o conhecimento da empresa nesta metodologia ágil.

v. Segunda Versão do Processo Padrão: As oportunidades de melhoria identificadas na execução do projeto piloto e a capacitação *Certified Scrum Masters* geraram novas sugestões de alteração no processo padrão. Foi elaborada, então, uma segunda versão do processo padrão, mais voltada para as práticas do Scrum e mais adaptada à realidade da empresa.

vi. Demais Projetos: O processo padrão revisto foi utilizado nos demais projetos da empresa, com o apoio de ferramentas que tornassem mais ágil a execução do processo e automatizassem algumas atividades.

2.6 Métricas

Uma métrica é a medição de um atributo (propriedades ou características) de uma determinada entidade (produto, processo ou recursos) e fornece uma parte importante dos dados necessários para administração de um projeto de software. (KOSCIANSKI, 2007).

Métricas de softwares nos possibilitam realizar uma das atividades mais fundamentais do processo de gerenciamento de projetos que é o planejamento. A partir deste, passamos a identificar a quantidade de esforço, o custo e as atividades que são necessárias para a realização do projeto.

Porém, do ponto de vista de medição de software elas são divididas em duas categorias: medidas diretas e indiretas. Podemos considerar como medidas diretas do processo de engenharia de software o custo e o esforço aplicados no desenvolvimento e manutenção do software e do produto, a quantidade de linhas de código produzidas e o total de defeitos registrados durante um determinado período de tempo. Todavia, a qualidade e a funcionalidade do software ou a sua capacidade de manutenção são mais difíceis de serem avaliadas e só podem ser medidas de forma indireta.

Na seção seguinte será abordado o método GQM, que tem o objetivo de facilitar a seleção de métricas para um projeto.

2.6.1 Método GQM

Segundo (KOSCIANSKI, 2007), GQM (Goal-Question-Metric), é um método para selecionar as métricas que são analisadas em determinado projeto, considerando seus objetivos.

No entanto o GQM estabelece a definição dos objetivos, os quais são estabelecidos de acordo com as necessidades do *stakeholders* e fixados em função dos requisitos, como também as questões que são definidas para realizar o trabalho de medição, estas são as perguntas que se espera responder com o estudo, e, por fim, define as métricas, as quais são determinadas de acordo com as categorias das informações que se deseja obter.

Como mostra a figura 6, pode ter um objetivo relacionado a um questionamento e obtendo uma métrica a ser avaliada, como também pode ter um objetivo ou vários relacionados a vários questionamentos que pode definir várias métricas, analisando o que deve ser medido representado de cima para baixo (objetivos – questionamentos - métricas), e como deve ser interpretado de baixo para cima (métricas – questionamentos – objetivo).

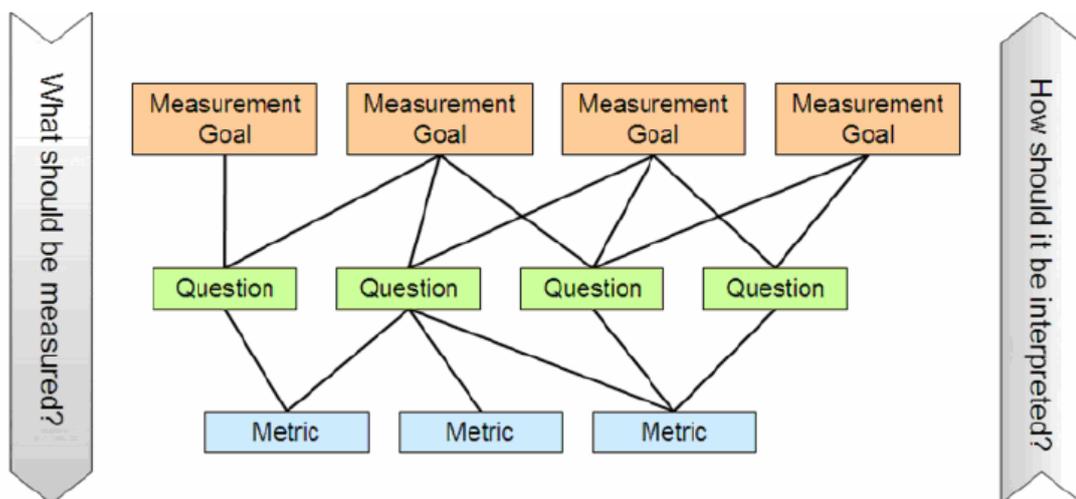


Figura 6 - Método GQM

Na próxima seção é abordada a métrica de dimensionamento de software, denominada Pontos de Função.

2.7 Pontos de Função

A métrica por função (Function Point – FP), inicialmente proposta por Albrecht, pode ser usada efetivamente como um meio para medir a funcionalidade entregue por um sistema. Usando dados históricos, o FP pode então ser usado para estimar o custo ou esforço necessário para projetar, codificar e testar o software, prever o número de erros que vão ser encontrados durante o teste; e prever o número de componentes e/ou o número de linhas de código projetadas no sistema implementado. São derivados usando uma relação empírica baseada em medidas de contagem (direta) do domínio de informação do software e avaliação da complexidade do software. (PRESSMAN, 2006)

No próximo capítulo será abordado o cenário da empresa que serviu de estudo de caso para realização desse trabalho, demonstrando o seu processo de desenvolvimento de software.

3- ESTUDO DE CASO

Este capítulo apresenta o estudo de caso da INFOX Tecnologia da Informação Ltda em busca da melhoria do processo atual, que atenda aos resultados esperados do nível G do MPS.BR e que tenha incorporado as mesmas técnicas e princípios do framework de desenvolvimento ágil Scrum, para o desenvolvimento dessa proposta se fez necessário conhecer a realidade da empresa que objetivasse alcançar o resultado pretendido através da aplicação das resoluções para os problemas encontrados durante a realização desse trabalho.

A organização deste capítulo é dada da seguinte forma: a Seção 3.1 descreve a empresa, inicialmente apresentando o negócio e o seu foco principal. A Seção 3.2, descreve o cenário atual encontrado no momento da realização desse trabalho. A Seção 3.3, descreve o PPDSI (Processo de Desenvolvimento de Software INFOX) e suas fases. A Seção 3.4, descreve os papéis existentes na empresa, e na Seção 3.5, demonstra os indicativos encontrados para os processos do nível G.

3.1 Conhecendo a Empresa

A INFOX Tecnologia da Informação Ltda é uma empresa especializada em desenvolvimento e integração de sistemas de informação em ambiente de fábrica de software e na prestação de serviços de suporte, fundada em 1986, com sede na cidade de Aracaju e escritório de representação nas cidades de Recife.

A empresa faz parte do CBTI (Consórcio Brasil de Tecnologia da Informação), adota sistema de gestão de qualidade ISO/IEC 9001:2008 e é fortemente comprometida com sistemas abertos, especialmente JAVA e LINUX, possuindo notória atuação em soluções de e-Gov. Conta com equipe dedicada à pesquisa, desenvolvimento e inovação, voltada para o estabelecimento de plataforma tecnológica que constitui a base para o desenvolvimento de aplicações corporativas com arquitetura MVC (*Model-View-Controll*).

3.2 Cenário Atual

A estrutura da empresa atualmente é composta pelos setores de Diretoria, P, D&I (Pesquisa, Desenvolvimento e Inovação), Qualidade, Escritório de Projetos, Tecnologia,

Suporte, Administrativo, Negócios, Análise, Desenvolvimento, SAC (Serviço de Atendimento ao Cliente), DBA (Administrador do Banco de Dados do inglês *DataBase Administrator*), Homologação e Documentação, conforme apresenta a Figura 5.

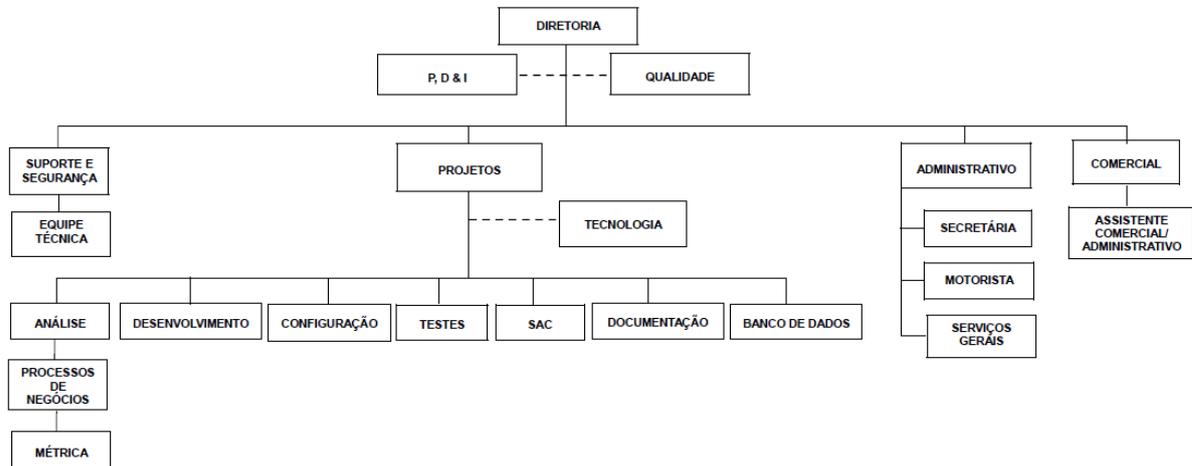


Figura 7 - Organograma da INFOX

O setor de Desenvolvimento denominado Fábrica de Software da INFOX explicado na Figura 6, mostrada a seguir, é composto atualmente por um diretor técnico, cinco gerentes de projetos, três gerentes de tecnologia, um gerente de qualidade, vinte e dois analistas, cinquenta e oito desenvolvedores, dois homologadores e dois documentadores (no total cento e vinte e sete profissionais, a maior parte com formação superior completa e especialização).

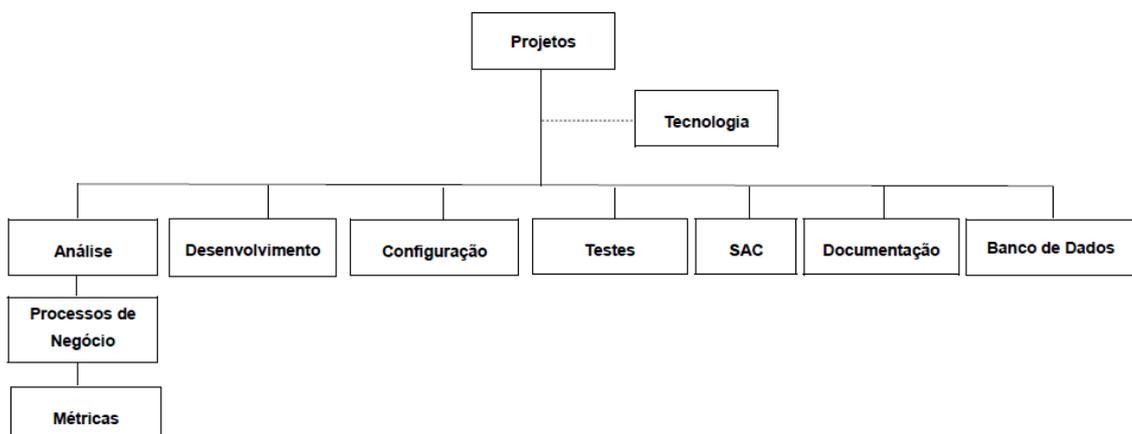


Figura 8 - Organograma da Fábrica de Software

A empresa através do apoio da alta direção e seus colaboradores obteve êxito na avaliação do modelo MPS.BR em maio de 2008, e em maio de 2011 a re-avaliação do

mesmo, onde conseguiu o nível G, esta avaliação foi inicialmente escolhido pela INFOX por ser um modelo genuinamente brasileiro e voltado para a realidade de pequenas e médias empresas.

Na seção seguinte, será abordado o processo de desenvolvimento de software da Infox, com o intuito de apresentar o mesmo e facilitar no entendimento, quando este é comparado com o processo desenvolvido.

3.3 Processo Padrão de Desenvolvimento de Software Infox

O PPDSI (Processo Padrão de Desenvolvimento de Software INFOX) descreve como os processos devem ser executados dentro do ciclo de vida de desenvolvimento de software na empresa, ele contém as fases, tarefas, definições dos papéis com habilidades e perfis, ferramentas utilizadas e toda documentação padronizada utilizada dentro do processo. A seguir são apresentadas cada fase do processo.

3.3.1 Fases

O ciclo de vida de desenvolvimento de software INFOX utilizado no processo padrão é o de cascata com iterações como mostra a Figura 7, por melhor se adequar à realidade dos projetos desenvolvidos na organização.

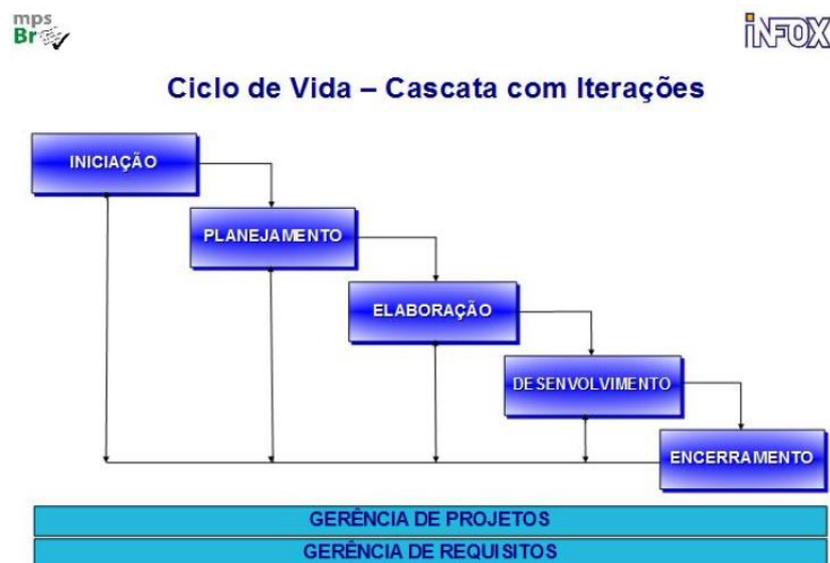


Figura 9 - Ciclo de Vida de Desenvolvimento

INICIAÇÃO - reconhecer que um projeto ou fase deve começar e se comprometer com a sua execução.

PLANEJAMENTO - planejar e manter um esquema de trabalho viável para atingir os objetivos de negócio que determinaram a existência do projeto.

ELABORAÇÃO - elaborar e validar todos os produtos de trabalho definidos para o projeto.

DESENVOLVIMENTO- coordenar pessoas e outros recursos para realizar o que foi planejado, assegurando que os objetivos do projeto estão sendo atingidos, através da monitoração e da avaliação do seu progresso, tomando ações corretivas quando necessárias.

ENCERRAMENTO- formalizar a aceitação do projeto ou fase e fazer a sua finalização, documentando as lições aprendidas.

Nas próximas seções será detalhado cada fase com seu fluxo de atividades, no qual as caixas em amarelo representam às atividades correspondentes a fase, o círculo verde o início do fluxo, e o vermelho o fim da fase, o *gateway* em laranja representa o fluxo de decisão.

3.3.1.1 Iniciação

Esta fase contempla os objetivos da iniciação de um projeto, com um fluxo contendo quatro atividades, como mostra a Figura 8.

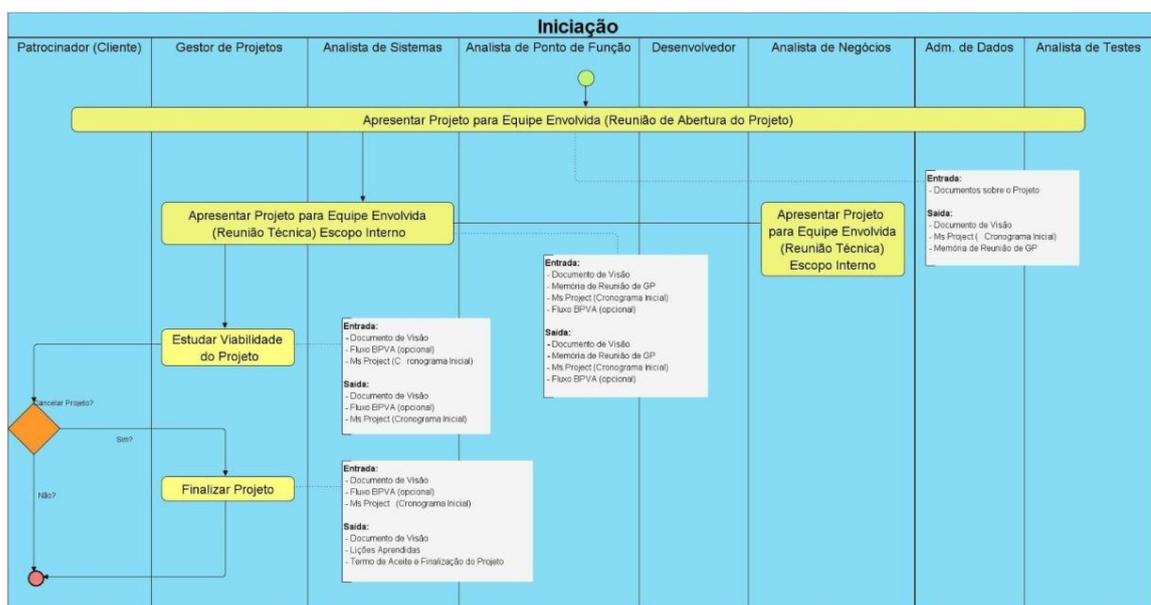


Figura 10 - Fase Iniciação

A seguir será detalhada cada atividade, com sua descrição e os papéis de quem realizam as mesmas.

3.3.1.1.1 Apresentar Projeto para Equipe Envolvida (Reunião de Abertura do Projeto)

Descrição: Reunião de Abertura do Projeto entre as partes envolvidas para uma primeira apresentação do escopo do projeto (visão geral) utilizando dados fornecidos pelo cliente como: descrição do projeto, objetivos mensuráveis ou justificativa do projeto, escopo do projeto, escopo do produto, requisitos das partes interessadas já conhecidos e risco de alto nível do projeto.

Partes Envolvidas: Patrocinador (cliente), Gestor de Projetos, Analista de Sistemas, Analista de Ponto de Função, Analista de Negócios, Desenvolvedor, Administrador de Dados e Analista de Testes.

3.3.1.1.2 Apresentar Projeto para Equipe Envolvida (Reunião Técnica) Escopo Interno

Descrição: Reunião Técnica entre o Gestor de projetos e o(s) Analista(s) para a definição final do escopo do projeto (visão geral) utilizando dados coletados no Cliente. Será definida a versão final do Documento de Visão com base nas informações coletadas na reunião de abertura do projeto. No Documento de Visão conterà as informações das pessoas envolvidas no projeto, objetivos mensuráveis, riscos preliminares, dentre outros. **Partes Envolvidas:** Gestor de Projetos, Analista de Negócios, Analista de Sistemas.

3.3.1.1.3 Estudar Viabilidade do Projeto

Descrição: Verificar se o projeto atende aos critérios de aceitação de viabilidade, levando em consideração fatores como custo, tempo de desenvolvimento, satisfação dos anseios do cliente, etc.

Parte Envolvida: Gestor de Projetos.

3.3.1.1.4 Finalizar Projeto

Descrição: Realizar o estudo da viabilidade do projeto utilizando os critérios de aceitação se o resultado dessa análise for negativo, deve ser realizada uma reunião para comunicar a todos os envolvidos a finalização do projeto em questão.

Parte Envolvida: Gestor de Projetos.

A fase seguinte do PPDSI é a de planejamento, que será apresentada na próxima seção.

3.3.1.2 Planejamento

Esta fase contempla os objetivos do planejamento de um projeto, com um fluxo de tarefas, como mostra a Figura 9.

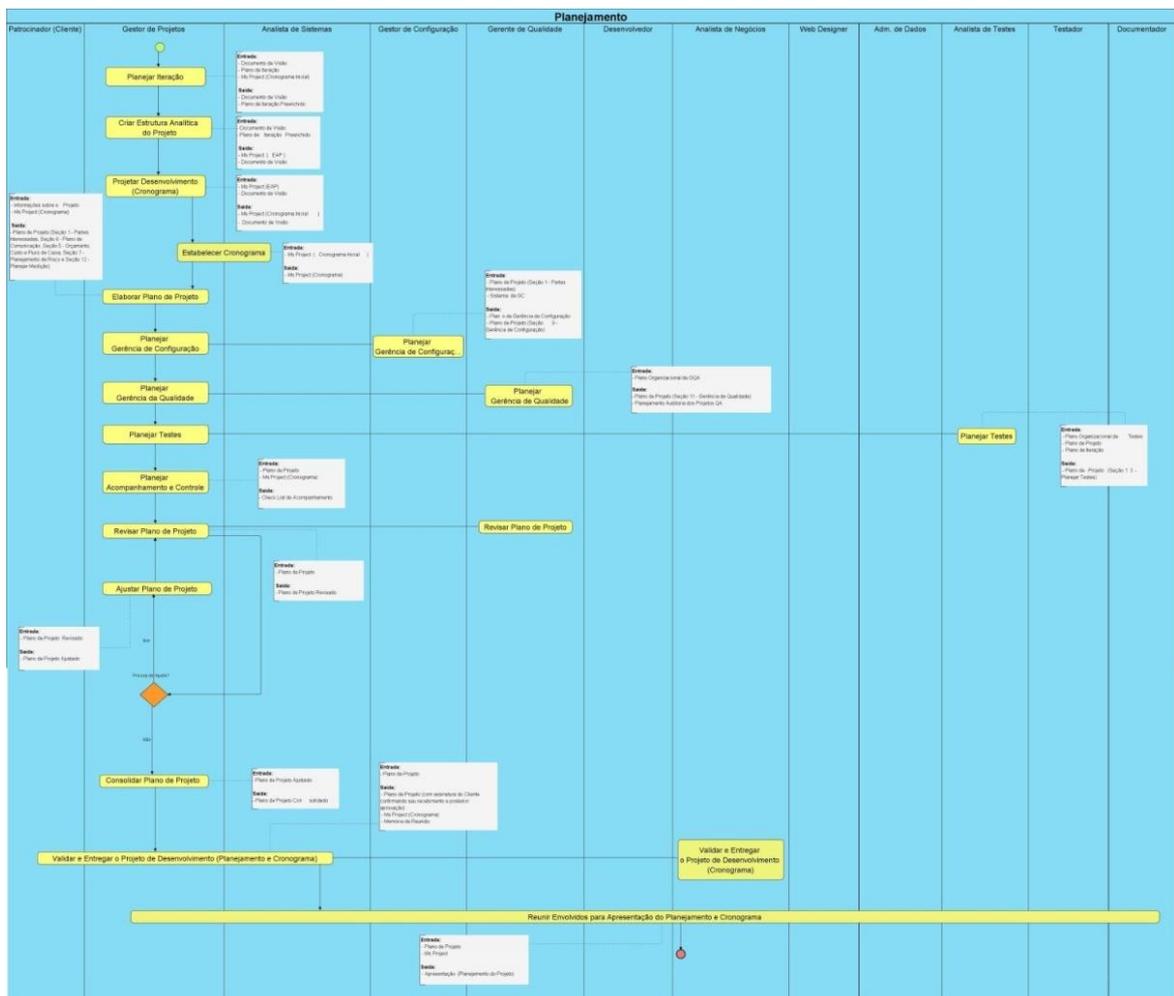


Figura 11 - Fase Planejamento

As atividades correspondentes a esta fase são detalhadas a seguir.

3.3.1.2.1 Planejar Iteração

Descrição: Definir um conjunto de iterações do projeto (módulos testáveis) e identificar os objetivos de cada uma dessas iterações.

Parte Envolvida: Gestor de Projetos.

3.3.1.2.2 Criar Estrutura Analítica do Projeto

Descrição: Definir a subdivisão do escopo do projeto com tarefas seqüenciadas, data de início e fim. É o processo de subdivisão das atividades, das entregas e de todo o trabalho necessário à execução do projeto em componentes menores e de gerenciamento mais fácil.

Parte Envolvida: Gestor de Projetos.

3.3.1.2.3 Projetar Desenvolvimento (Cronograma)

Descrição: Elaboração do cronograma do projeto com base na subdivisão das atividades e entregas definidas.

Partes Envolvidas: Gestor de Projetos.

3.3.1.2.4 Estabelecer Cronograma

Descrição: Estabilizar o cronograma definido para execução das atividades do projeto. Um dos principais fatores de sincronicidade das etapas de um projeto é o estabelecimento do calendário para execução das tarefas validado com o cliente.

Parte Envolvida: Gestor de Projetos, Analista de Sistemas.

3.3.1.2.5 Elaborar Plano de Projeto

Descrição: Reunir as informações sobre o projeto, documentar o resultado da análise de viabilidade observada no documento de visão e EAP - Estrutura Analítica do Projeto. Identificar os recursos humanos que são alocados ao projeto. Fazer um plano de comunicação entre os envolvidos do projeto, a avaliação dos custos para o projeto, o

levantamento dos possíveis riscos que poderão acontecer no decorrer da sua execução, sinalizando no documento o grau de probabilidade, impacto e prioridade. Planejar a análise e as métricas para o projeto e processo.

Parte Envolvida: Gestor de Projetos.

3.3.1.2.6 Planejar a Gerência de Configuração

Descrição: Planejar o versionamento, solicitação de mudanças e integração contínua do projeto.

Partes Envolvidas: Gestor de Configuração, Gestor de Projetos.

3.3.1.2.7 Planejar a Gerência de Qualidade

Descrição: Planejar as auditorias físicas e funcionais durante toda a execução do projeto, assegurando que o processo definido para o projeto seja estabelecido e institucionalizado.

Parte Envolvida: Gerente de Qualidade, Gestor de Projetos.

3.3.1.2.8 Planejar Testes

Descrição: Planejar os testes que serão realizados durante toda a execução do projeto, assegurando que o processo definido para os testes do projeto seja estabelecido e institucionalizado.

Partes Envolvidas: Analista de Testes, Gestor de Projetos.

3.3.1.2.9 Planejar Acompanhamento e Controle

Descrição: Elaborar todo o acompanhamento e controle que será realizado no projeto durante sua execução.

Parte Envolvida: Gestor de Projetos.

3.3.1.2.10 Revisar Plano de Projeto

Descrição: Avaliar e revisar o plano de projeto, assinalando os possíveis erros encontrados.

Partes Envolvidas: Gestor de Projetos, Gerente de Qualidade.

3.3.1.2.11 Ajustar Plano de Projeto

Descrição: São feitos os ajustes necessários verificados na revisão do documento.

Parte envolvida: Gestor de Projetos.

3.3.1.2.12 Consolidar Plano de Projeto

Descrição: Tornar sólida as informações reunidas no plano de projeto em relação aos Requisitos validados para o projeto.

Parte Envolvida: Gestor de Projetos.

3.3.1.2.12 Validar e Entregar o de Projeto (Cronograma)

Descrição: Nesta tarefa, o plano de projeto e o cronograma serão entregues ao cliente.

Partes Envolvidas: Patrocinador (cliente), Gestor de Projetos, Analista de Sistemas, Analista de Negócios.

3.3.1.2.13 Reunir Envolvidos para Apresentação do Planejamento

Descrição: Ao término do planejamento aprovado pelo cliente deve ser realizada uma reunião com todos os membros envolvidos onde é feita uma apresentação de todos os itens que foram validados com o cliente. É importante destacar quais as justificativas, restrições e demais itens relevantes.

Partes Envolvidas: Patrocinador (cliente), Gestor de Projetos, Analista de Sistemas, Analista de Negócios, Gestor de Configuração, Gerente de Qualidade, Desenvolvedor, Web Designer, Administrador de Dados, Analista de Testes, Testador, Documentador.

Esta fase contemplou as atividades referentes ao planejamento de um projeto, e subsidia a execução das próximas fases.

3.3.1.3 Elaboração

Esta fase contempla a elaboração e validação de todos os produtos de trabalho definidos para o projeto, como mostra a Figura 10.

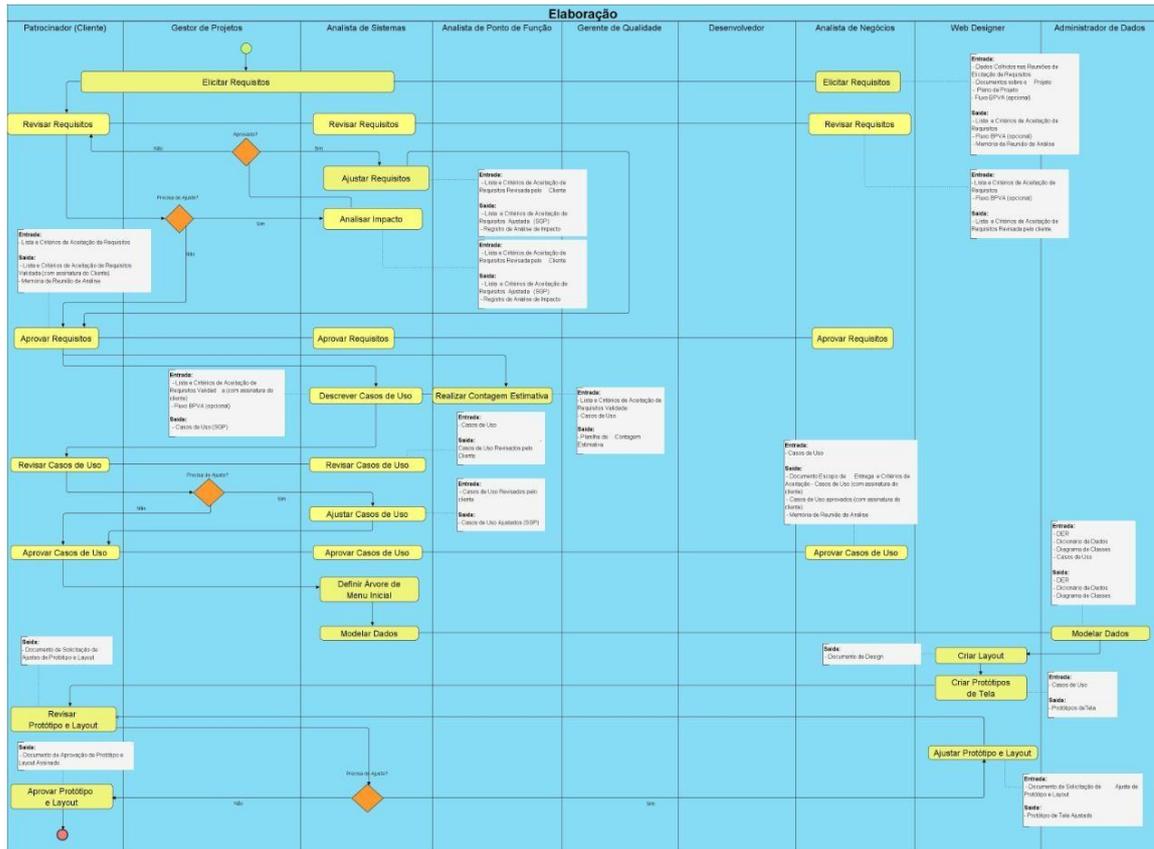


Figura 12 - Fase Elaboração

A seguir é detalhada cada atividade, com a respectiva descrição e os papéis de quem a executa.

3.3.1.3.1 Elicitar Requisitos

Descrição: Definir com o Cliente quem será, de sua parte, o responsável pelo Projeto (Patrocinador do Projeto) e o fornecedor de requisitos. Essa informação estará no documento Plano de Projeto na seção Partes Interessadas. Após a definição do patrocinador e fornecedor de requisitos, deverá ser cadastrado no Sistema Gerenciador de Processos (SGP).

Partes Envolvidas: Patrocinador (cliente), Gestor de Projetos Analista de Negócios Analista de Sistemas.

3.3.1.3.2 Revisar Requisitos

Descrição: O objetivo da revisão da elicitação de requisitos é identificar as possíveis mudanças nos requisitos do cliente. A revisão é de suma importância para avaliarmos se o que está sendo elaborado está consistente com as necessidades do cliente. O cliente fará uma revisão, estando os requisitos revisados e aprovados o processo de desenvolvimento continuará.

Partes Envolvidas: Patrocinador (cliente), Analista de Sistemas, Analista de Negócios.

3.3.1.3.3 Ajustar Requisitos

Descrição: Alteração necessária solicitada pelo cliente no requisito elicitado.

Parte Envolvida: Analista de Sistemas.

3.3.1.3.4 Analisar Impacto

Descrição: Realizar a análise de Impacto proveniente da alteração necessária solicitada pelo cliente no requisito elicitado.

Parte Envolvida: Analista de Sistemas.

3.3.1.3.5 Aprovar Requisitos

Descrição: Aprovar os requisitos elicitados que estão em conformidade e alinhados com os critérios de aceitação.

Partes Envolvidas: Patrocinador (cliente), Analista de Sistemas, Analista de Negócios.

3.3.1.3.6 Descrever Casos de Uso

Descrição: Nesta atividade, serão elaborados os Casos de Uso tendo como base os requisitos definidos na fase anterior. Os Casos de Uso deverão ser cadastrados no Sistema

Gerenciador de Projetos (SGP). O Caso de Uso validado e aprovado pelo Cliente irá gerar, na fase de desenvolvimento, uma tarefa para o desenvolvedor.

Parte Envolvida: Analista de Sistemas.

3.3.1.3.7 Realizar Contagem Estimativa

Descrição: Realizar a contagem estimativa dos requisitos e casos de uso aprovados pelo cliente.

Parte Envolvida: Analista de Ponto de Função.

3.3.1.3.8 Revisar Casos de Uso

Descrição: Antes da revisão feita pelo cliente, deve ser realizada uma revisão por outro analista que não tenha participado da descrição dos casos de uso colocando em prática a revisão por pares. O Cliente deve fazer a revisão dos Casos de Uso entregues, verificando se eles atendem ao que foi solicitado para o projeto. As anotações com os ajustes necessários serão feitas no próprio Caso de Uso.

Possíveis Fluxos:

- Caso não haja ajustes a serem feitos, a tarefa Ajustar Casos de Uso será ignorada, passando o fluxo para a tarefa Aprovar Casos de Uso.
- Se for detectada a necessidade de alterações, deverá solicitá-las no próprio Caso de Uso (verso) e o fluxo passa para a tarefa - Ajustar Casos de Uso. Após esses ajustes, o fluxo volta para Revisar Casos de Uso.

Partes Envolvidas: Patrocinador (cliente), Analista de Sistemas.

3.3.1.3.9 Ajustar Casos de Uso

Descrição: Se detectada pelo Cliente a necessidade de alterações nos Casos de Uso, ou seja, se alguma parte do escopo do produto não está atendendo as suas requisições, o fluxo entra nessa tarefa para os devidos ajustes.

Parte Envolvida: Analista de Sistemas.

3.3.1.3.10 Aprovar Casos de Uso

Descrição: Após a revisão dos Casos de Uso, bem como dos ajustes que tenham sido realizados (caso necessário), o cliente deverá dar a aprovação final do Projeto Lógico, o que será formalizado através do documento específico.

Partes Envolvidas: Patrocinador (cliente), Analista de Sistemas, Analista de Negócios.

3.3.1.3.11 Definir Árvore de Menu Inicial

Descrição: O analista deverá definir os menus utilizados inicialmente.

Parte Envolvida: Analista de Sistemas.

3.3.1.3.12 Modelar Dados

Descrição: O Administrador de Dados irá analisar o DER e o Dicionário de Dados juntamente com os Casos de Uso e fará os ajustes necessários, gerando o modelo de dados final. Logo após a elaboração e validação dos Casos de Uso, é feita a definição da modelagem de dados. O analista realiza as possíveis melhorias nas tabelas definidas dessa tarefa, sairá o DER e o dicionário de dados.

Parte Envolvida: Analista de Sistemas, Administrador de Dados.

3.3.1.3.13 Criar Layout

Descrição: Nesta tarefa, o responsável pelo design irá elaborar o layout do sistema com base nas informações obtidas na tarefa anterior. É preciso que o layout seja um elo de comunicação com o usuário, que sua linguagem seja condizente com o objetivo proposto. Utilizar heurísticas de usabilidade fundamental para se gerar layouts para a web.

Parte Envolvida: Web Designer.

3.3.1.3.14 Criar Protótipos de Telas

Descrição: O Designer e o desenvolvedor devem criar as telas de acordo com a árvore de menu definida.

Partes Envolvidas: Desenvolvedor, Web Designer.

3.3.1.3.15 Revisar Protótipos e Layout

Descrição: O Cliente fará as devidas verificações no layout apresentado, confrontando-o com as informações do documento de Design e verificando se realmente o que foi realizado atende suas expectativas.

Possíveis Fluxos:

- Caso o cliente aprove o layout desenvolvido, a tarefa Ajustar Protótipo e Layout será ignorada e o fluxo passará para a tarefa Aprovar do Protótipo e Layout.
- Se forem necessárias alterações, deverá preencher o documento adequado para essas solicitações e passa para a atividade Ajustar Protótipo e Layout. Após esses ajustes, o fluxo volta para a tarefa Revisar Protótipo e Layout para que seja feita nova verificação.

Parte Envolvida: Patrocinador (cliente).

3.3.1.3.16 Ajustar Protótipo e Layout

Descrição: As partes envolvidas farão os ajustes solicitados pelo Cliente com base no documento enviado pelo mesmo.

Parte Envolvida: Web Designer.

3.3.1.3.17 Aprovar Protótipo e Layout

Descrição: Após homologado layout e protótipo bem como todos os ajustes que tenham sido realizados (caso necessário), o cliente deverá fazer a aprovação final que será formalizada através de documento específico.

Parte Envolvida: Patrocinador (cliente).

Esta fase contemplou as atividades correspondentes a elaboração de um projeto e os artefatos gerados pela mesma são essenciais para a execução da próxima fase.

3.3.1.4 Desenvolvimento

Esta fase contempla o desenvolvimento de um projeto, assegurando que os objetivos do projeto estão sendo atingidos, através da monitoração e da avaliação do seu progresso, como mostra a Figura 11.

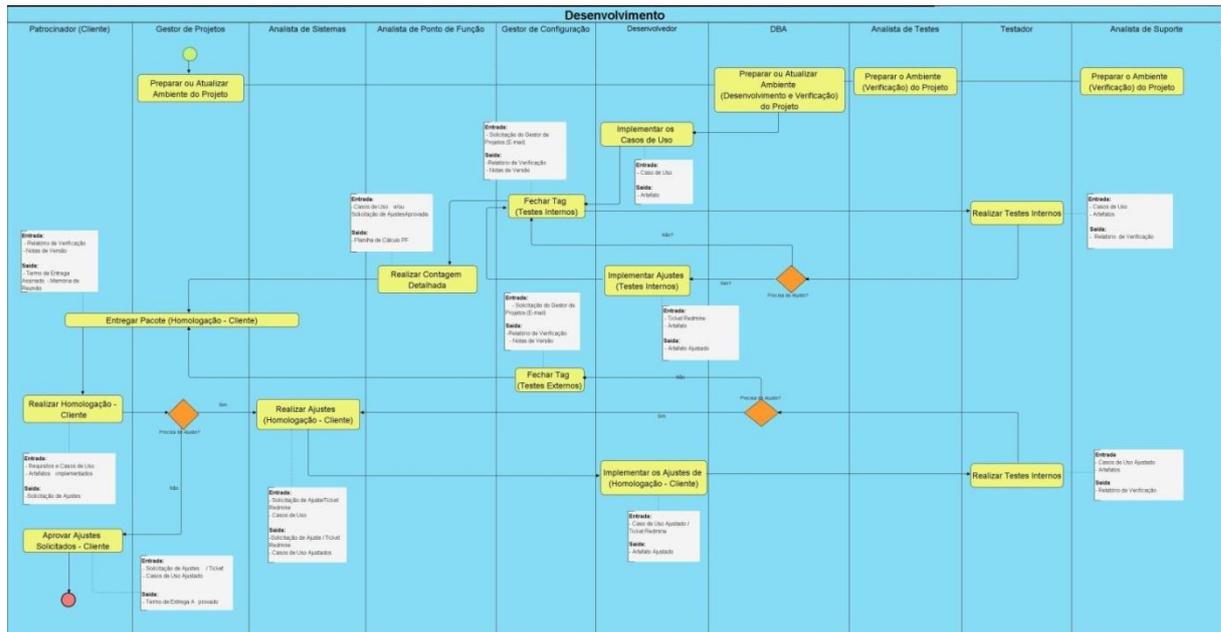


Figura 13 - Fase – Desenvolvimento

3.3.1.4.1 Preparar ou Atualizar os Ambientes (Desenvolvimento e Verificação) do Projeto

Descrição: Preparação ou atualização dos ambientes (Desenvolvimento e Testes-verificação). Para a criação do ambiente do projeto deve-se levar em consideração o Manual de Padronização de Banco de Dados ou Manual de Padronização de Classes que contém toda a padronização para o banco de dados, modelagem, nomenclaturas, dentre outras. Após a criação do ambiente, é feito um *checklist* de aderência da padronização ao projeto.

Partes Envolvidas: Gestor de Projetos, DBA (Administrador de Banco de Dados) Analista de Testes, Analista de Suporte.

3.3.4.2 Implementar Casos de Uso

Descrição: Nesta tarefa, o desenvolvedor irá implementar os Casos de Uso (artefato) passados pelo analista. Isso será feito através de tarefas a serem cadastradas no Sistema Gerenciador de Processos (SGP).

Parte Envolvida: Desenvolvedor.

3.3.1.4.3 Fechar Tag (Testes Internos)

Descrição: Nesta tarefa, a pessoa responsável pela área de configuração mediante autorização do gestor de projetos irá fechar uma *tag* I, contendo o escopo definido para entrega, onde serão realizados os testes internos.

Parte Envolvida: Gestor de Configuração.

3.3.1.4.4 Realizar Testes Internos

Descrição: Nesta tarefa, a pessoa responsável pela homologação fará os testes necessários, de acordo com o acordado com o gestor de projetos, para verificar se o que foi implementado está atendendo aos requisitos e aos casos de uso definidos para o projeto. É nessa tarefa que se deve identificar as "falhas" de desenvolvimento do sistema, para que estas sejam retificadas antes da entrega do software, ou nova versão, para o cliente.

Parte Envolvida: Testador.

3.3.1.4.5 Implementar Ajustes (Testes Internos)

Descrição: O desenvolvedor irá realizar os ajustes no artefato conforme as anotações descritas pelo testador no *ticket* da ferramenta *Redmine*, podendo ocorrer versionamento no Caso de Uso.

Parte Envolvida: Desenvolvedor.

3.3.1.4.6 Realizar Contagem Detalhada

Descrição: Realizar a contagem detalhada dos artefatos implementados e validados pelo cliente.

Parte Envolvida: Analista de Ponto de Função.

3.3.1.4.7 Entregar Pacote (Homologação – Cliente)

Descrição: Será feita a entrega ao cliente do Relatório de Verificação com orientações sobre as funcionalidades testadas no pacote (escopo definido). Para realização da

homologação externa é preciso preparar o Banco de Dados de Teste contendo novos dados ou com dados que possuam garantia de integridade. A *tag* H será disponibilizada no ambiente do cliente, o cliente deverá assinar F-139_Termo_de_Entrega confirmando que recebeu os artefatos para devida homologação.

Parte Envolvida: Patrocinador (cliente), Gestor de Projetos, Analista de Sistemas.

3.3.1.4.8 Realizar Homologação – Cliente

Descrição: O cliente irá realizar os testes de homologação para verificar se a implementação corresponde ao que está descrito nos requisitos e casos de uso (que foram validados por ele na fase de Planejamento), e se realmente o que foi implementado corresponde as suas expectativas sobre o projeto.

Possíveis Fluxos:

- Caso o cliente encontre algum erro deverá preencher o documento de Solicitação de Ajuste e o fluxo passa para a atividade Realizar Ajustes (Homologação - Cliente).
- Se for não forem encontrados erros, o fluxo passa diretamente para a atividade Aprovar

Parte Envolvida: Patrocinador (cliente).

3.3.1.4.9 Realizar Ajustes (Homologação – Cliente)

Descrição: Ajustes no sistema conforme anotações feitas pelo cliente no Documento de Solicitação de Ajuste. Após realizados os ajustes, eles devem ser novamente validados pelo Testador (atividade: Realizar Testes Internos). Só após essa validação retorna para a homologação feita pelo cliente (atividade: Entregar Pacote para Homologação - Cliente). Os ajustes devem ser classificados como: bug, ajustes no escopo ou nova funcionalidade. Caso o ajuste seja classificado como bug, deve ser reportado no Redmine. Caso o ajuste seja classificado como ajuste no escopo, o caso de uso será versionado e serão realizados novos testes utilizando o documento Relatório de Verificação.

Parte Envolvida: Analista de Sistemas.

3.3.1.4.10 Implementar Ajustes (Homologação – Cliente)

Descrição: O desenvolvedor irá realizar os ajustes no artefato conforme as modificações realizadas no Caso de Uso através do versionamento ou a descrição do ticket do *Redmine*.

Parte Envolvida: Desenvolvedor.

3.3.1.4.11 Aprovar Ajustes Solicitados – Cliente

Descrição: Após realizada a homologação dos artefatos e dos ajustes feito pelo desenvolvedor, o Cliente dará a aprovação final de cada Caso de Uso relacionada com a solicitação de ajuste (caso exista), onde deverá assinar o documento.

Parte Envolvida: Patrocinador (cliente).

Como última fase correspondente ao processo, temos a de encerramento, que será apresentada na próxima seção.

3.3.1.5 Encerramento

Esta fase contempla o encerramento de um projeto, com seu fluxo de tarefas demonstrado na Figura 12.

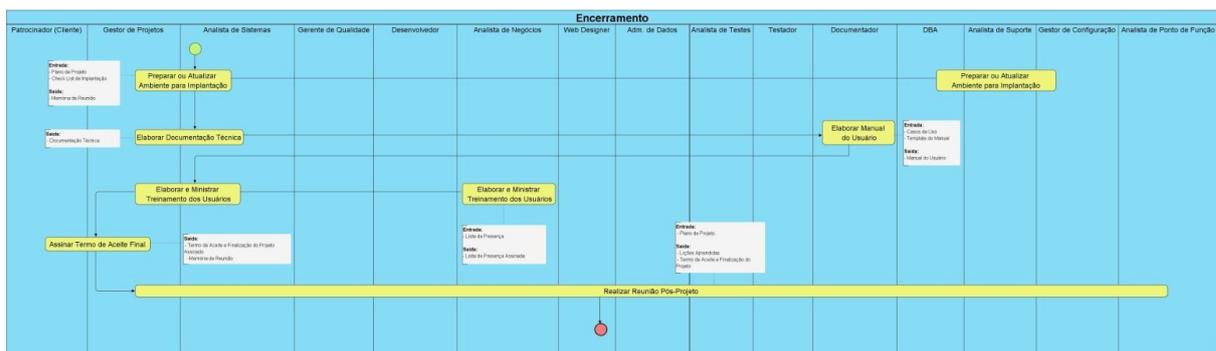


Figura 14 - Fase – Encerramento

3.3.1.5.1 Preparar ou Atualizar Ambiente para Implantação

Descrição: Preparação ou atualização do ambiente para implantação. Trazer os dados originais da base do cliente e adaptá-lo a realidade da nova aplicação.

Partes Envolvidas: Gestor de Projetos, Analista de Sistemas, DBA (Administrador de Banco de Dados), Analista de Suporte, Gestor de Configuração.

3.3.1.5.2 Elaborar Documentação Técnica

Descrição: Será elaborada toda a documentação técnica do Projeto que deverá ser entregue ao Cliente. Alguns itens que devem constar na documentação:

- Scripts para geração das Entidades;
- Diagrama de Entidades e Relacionamentos (DER);
- Diagrama de Classes;
- Documentação das regras de negócio.

Partes Envolvidas: Gestor de Projetos, Analista de Sistemas.

3.3.1.5.3 Elaborar Manual do Usuário

Descrição: Elaboração do Manual do Usuário.

Parte Envolvida: Documentador.

3.3.1.5.4 Elaborar e Ministrare Treinamento dos Usuários

Descrição: O analista de negócios (conhecedor da regra de negócio) deverá montar a estrutura do curso de treinamento. Ele deve elaborar um conteúdo programático contendo os tópicos a ser ministrado, o treinamento será ministrado pelo analista ou gestor de projetos a depender da disponibilidade. Essa tarefa pode ser executada pelo cliente, caso ele tenha disponibilidade e preferência em ministrar o treinamento para os usuários finais, nesse caso deve ser informado que essa tarefa ficará sobre a responsabilidade do cliente no plano de projeto.

Partes Envolvidas: Gestor de Projetos, Analista de Sistemas, Analista de Negócios.

3.3.1.5.5 Assinar Termo de Aceite Final

Descrição: O Cliente assina dando o aceite final de que o objeto do contrato foi finalizado e entregue de acordo com suas perspectivas.

Partes Envolvidas: Patrocinador (cliente), Gestor de Projetos.

3.3.1.5.6 Realizar Reunião Pós-Projeto

Descrição: Ao término do projeto é realizada uma reunião com todos os membros envolvidos onde é feita uma retrospectiva de todo o processo realizado. É importante destacar quais os pontos mais positivos, negativos e as lições aprendidas para que os procedimentos estejam sempre em constante melhoria.

Partes Envolvidas: Gestor de Projetos, Analista de Negócios, Analista de Sistemas, Analista de Ponto de Função, Analista de Testes, Analista de Suporte, Gerente de Qualidade, Gestor de Configuração, Desenvolvedor, Web Designer, DBA (Administrador de Banco de Dados), Administrador de Dados, Testador / Documentador.

Na próxima seção serão detalhados os papéis existentes no PPDSI.

3.3.2 Papéis do PPDSI

Gestor de projetos- Disponibiliza as informações necessárias para o andamento do projeto. Desenvolve soluções de administração de projetos que viabilizem a implementação no prazo e dentro do orçamento, alinhadas com as exigências de qualidade da Infox.

Analista - Realiza o levantamento de escopo do software e apóia a área de negócios na elaboração da proposta comercial, realiza a parte de especificação, orienta os desenvolvedores na implementação, lidera homologação interna, registra realizações das tarefas no SGP (Sistema Gestor de Projetos).

Desenvolvedor - Implementa a especificação definida nos prazos planejados, tendo como meta entregar o produto final com falha ZERO, realiza testes unitários e no browser, utilizar com rigor os padrões de desenvolvimento da fábrica, registra a realização das tarefas no SGP (Sistema Gestor de Projetos).

Web Designer - Mantém o website da fábrica e de seus projetos, cria projeto gráfico para os projetos da fábrica, cria novos gráficos para os componentes da fábrica.

DBA (Administrador de Banco de Dados) - Instala, configura e monitora o banco, verifica o uso de memória, verifica o tamanho do disco, verifica os logs do banco. Instala, migra e aplica *PATCH*, atualiza as versões do banco de dados garantindo integridade,

segurança, disponibilidade e performance. Faz *backup* e *restore*, realiza atualizações corretivas e evolutivas, monta e configura o ambiente de replicação. Domínio em *Tuning* de Banco, domínio completo na administração de servidores, registra a realização das tarefas no SGP.

AD (Administrador de Dados) - Mantém o DER, trabalha com normalização, soluções para melhoramento de performance, soluções para maximizar o compartilhamento de dados entre as aplicações, minimiza redundância, elimina inconsistência, mantém a padronização de nomenclatura dos objetos, mantém a padronização da estrutura da base de dados, extração de dados, replicação de dados entre as bases através do Java ou ferramenta externa.

Testador e Documentador - Realiza a homologação interna (*checklist* dos padrões da fábrica), realiza os testes em diversos browsers, elabora o manual da aplicação (do usuário), verifica a utilização dos padrões de desenvolvimento da fábrica, sinaliza aos responsáveis a não utilização desses padrões, registra a realização das tarefas no SGP.

Suporte ao usuário - Atende as solicitações do usuário pelo telefone, via e-mail ou em domicílio e soluciona os problemas.

3.4 Indicativos Encontrados nos Processos

Como parte da fase de análise das práticas, processos e procedimentos da empresa, definido para elaboração deste trabalho, como é mostrado no capítulo 4 seção 4.1, foi realizado uma avaliação dos resultados esperados para os processos do nível G encontrados na Infox. Definindo o indicativo encontrado em três graus: bom, satisfatório, e insatisfatório. A seção 3.3.1, demonstra os indicativos encontrados relacionados à Gerência de Requisitos. A seção 3.3.2 demonstra os indicativos encontrados relacionados à Gerência de Projetos.

3.4.1 Gerência de Requisitos

A tabela apresentada na figura demonstra os indicativos encontrados para o processo de Gerência de Requisitos (GRE) verificados na INFOX, levando em consideração os Atributos de Processo (AP) e seus resultados (RAP). Estes indicativos foram obtidos através de entrevistas com a gerente de qualidade da empresa.

Tabela 1 - Indicativos encontrados para o processo de Gerência de Requisitos

Código	Descrição do Processo	Indicativo Encontrado
GRE1	Os requisitos são entendidos, avaliados e aceitos juntos aos fornecedores de requisitos, utilizando critérios objetivos.	Bom
GRE 2	O comprometimento da equipe técnica com os requisitos aprovado é obtido.	Satisfatório
GRE 3	A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida.	Bom
GRE 4	Revisões em planos e produtos de trabalho do projeto são realizados visando a identificar e corrigir inconsistências em relação aos requisitos.	Satisfatório
GRE 5	Mudanças nos requisitos são gerenciados ao longo do projeto.	Bom

3.4.2 Gerência de Projetos

A tabela a seguir demonstra os indicativos encontrados para o processo de Gerência de Projetos (GPR) verificados na INFOX, levando em consideração os Atributos de Processo (AP) e seus resultados (RAP). Estes indicativos foram obtidos através de entrevistas com o gerente de portfólio da empresa.

Tabela 2 - Indicativos encontrados para o processo de Gerência de Projetos

Código	Descrição do Processo	Indicativo Encontrado
GPR1	O escopo do trabalho para o projeto é definido.	Bom
GPR2	As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados.	Satisfatório

GPR3	O modelo e as fases do ciclo de vida do projeto são definidos.	Satisfatório
GPR4	O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas.	Bom
GPR5	O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos.	Bom
GPR6	Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são denominados e identificados.	Bom
GPR7	Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo.	Bom
GPR8	Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados.	Bom
GPR9	Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança.	Satisfatório
GPR10	Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos.	Bom
GPR11	A viabilidade de atingir as metas do projeto, considerando as restrições e	Bom

	os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados.	
GPR12	O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido.	Bom
GPR13	O projeto é gerenciado utilizando-se o Plano do Projeto e outros planos que afetam o projeto e os resultados são documentados.	Bom
GPR14	O envolvimento das partes interessadas no projeto é gerenciada.	Bom
GPR15	Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento.	Bom
GPR16	Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas.	Bom
GPR17	Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.	Bom

4- METODOLOGIA DE IMPLEMENTAÇÃO DO PROCESSO

Neste capítulo apresentamos a metodologia utilizada para implementação do processo, a qual servirá de base para entender os passos ou fases da elaboração deste trabalho. A organização do mesmo é dada da seguinte forma, a Seção 4.1 descreve as fases da estratégia de implementação, a Seção 4.2 descreve as fases do processo concebido com seu sub-processos e a Seção 4.3 descreve as métricas utilizadas no estudo de caso.

4.1- Fases da Estratégia de Implementação

A definição do processo do nível G do MPS.BR baseado no Scrum para a INFOX, foi determinado seguindo a estratégia de implementação demonstrada na Figura 13, dividida em cinco etapas, que utilizou como base a estratégia para implementação de processos apresentada na Seção 2.5.

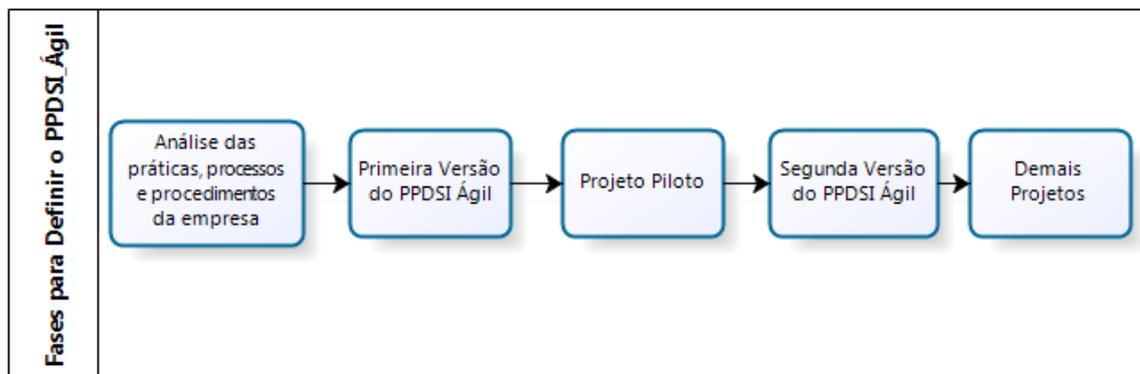


Figura 15 - Fases da Estratégia de Implementação

A seguir é detalhada cada etapa:

* *Análise das práticas, processos e procedimentos da empresa*: As práticas, os processos e procedimentos já utilizados pela empresa foram analisados, com o objetivo de identificar e entender as necessidades da empresa, permitindo a elaboração de um processo padrão ágil que atendesse a estas necessidades.

* *Primeira Versão do PPDSI Ágil*: Foi elaborada a primeira versão do processo, denominada PPDSI Ágil, seguindo o nome padrão do processo da empresa e adicionado da palavra ágil para representar a adoção do Scrum. Este buscou atender às necessidades

identificadas na fase anterior. Durante todo o ciclo de vida de elaboração do processo, foram feitas várias validações com a representante da empresa, mostrando e discutindo tudo o que tava sendo feito. O objetivo desta primeira versão é permitir que a empresa utilize o processo rapidamente em um projeto piloto e aprenda com esta experiência, evoluindo o processo e adequando-o à realidade da empresa.

* *Projeto Piloto*: O PPDSI Ágil foi utilizado em um primeiro projeto, com o objetivo de identificar várias oportunidades de melhoria no processo.

* *Segunda Versão do Processo Padrão*: Depois de executado o processo no projeto piloto, foram observadas as oportunidades de melhoria que resultou em alteração no processo.

* *Demais Projetos*: O processo padrão revisto para ser utilizado (institucionalizado) nos demais projetos da empresa.

Na próxima seção são abordadas as fases do processo desenvolvido, que foram utilizadas na aplicação do mesmo.

4.2- Fases do Processo Concebido

O processo concebido foi denominado como PPDSI_Ágil e está dividido em três grandes fases, baseado no modelo Scrum (Pré-Jogo, Jogo e Pós-Jogo) como citado na Seção 2.4.4. As fases do novo processo ficaram definidas como Pré-Execução, Execução e Encerramento, no qual dentro de cada fase é especificado o fluxo de atividades, com papéis e artefatos. A Figura 14 representa o processo definido com as respectivas fases, cada fase é considerada um sub-processo, e o sub-processo de execução é iterativo, ou seja, acontecem várias vezes.

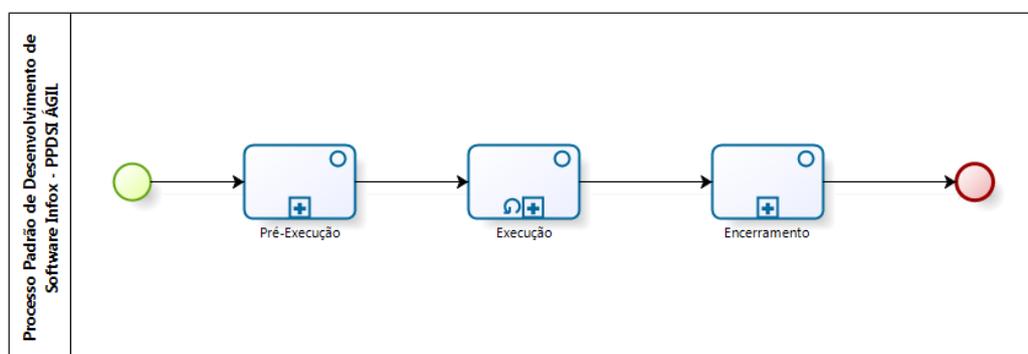


Figura 16 - Fases do Processo Concebido

Nas próximas seções são abordadas as fases do processo concebido, detalhando-as individualmente.

4.2.1- Pré-Execução

Esta fase contempla as atividades que são executadas antes de o projeto iniciar, obtendo o levantamento das informações iniciais, bem como a análise dos requisitos do produto, para o planejamento do projeto. Como artefatos desta fase terão a visão do produto, ou planejamento macro, o estudo de viabilidade, o *product backlog* inicial, sendo que este irá evoluir a na fase de execução, como também definir a formação do time, e caso ocorra à finalização do projeto irá gerar o termo de finalização do projeto e as lições aprendidas.

O fluxo de atividades desta fase é representado como mostra a Figura 15, no qual as caixas em azul representam as atividades, o círculo verde o início da fase e o vermelho o final, neste caso temos dois círculos vermelho, pois depois de realizado o estudo da viabilidade do projeto e verificado alguns critérios que impedem a execução do mesmo este será finalizado. Os *gateways* em amarelo representam o fluxo de decisão.

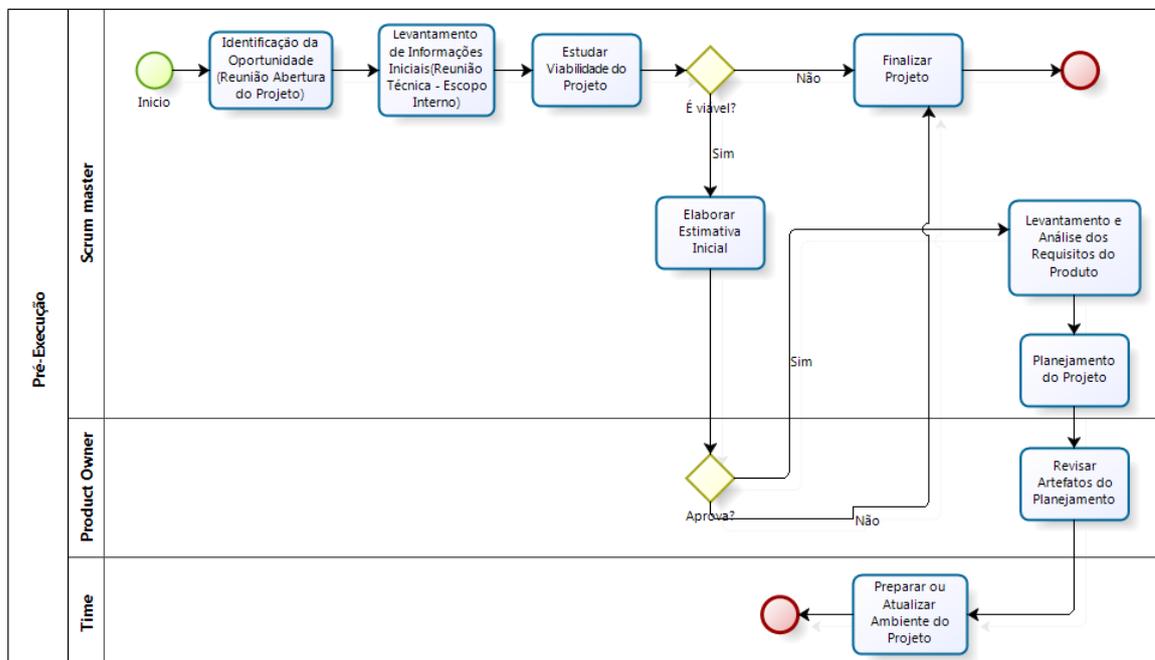


Figura 17 - Fluxo de Atividades da Fase de Pré - Execução

A seguir será detalhada cada atividade, com sua descrição e os papéis de quem realizam as mesmas.

4.2.1.1 Identificação da Oportunidade

Descrição: Reunião de Abertura do Projeto entre as partes envolvidas para uma primeira apresentação do escopo do projeto (visão geral) utilizando dados fornecidos pelo cliente.

Partes Envolvidas: *Scrum Master e Product Owner.*

4.2.1.2 Levantamento de Informações Iniciais

Descrição: Será definido o Documento de Visão com base nas informações coletadas na reunião de abertura do projeto. Neste documento conterà as informações das pessoas envolvidas no projeto, objetivos mensuráveis, riscos preliminares, dentre outros.

Partes Envolvidas: *Scrum Master e Product Owner.*

4.2.1.3 Estudar Viabilidade do Projeto

Descrição: Verificar se o projeto atende aos critérios de aceitação de viabilidade, levando em consideração fatores como custo, tempo de desenvolvimento, satisfação dos anseios do cliente, etc.

Parte Envolvida: *Scrum Master e Product Owner.*

4.2.1.4 Finalizar Projeto

Descrição: Depois de realizado o estudo da viabilidade do projeto e se o resultado dessa análise for negativo, deve ser realizada uma reunião para comunicar a todos os envolvidos a finalização do projeto em questão.

Parte Envolvida: *Scrum Master.*

4.2.1.5 Levantamento e Análise dos Requisitos do Produto

Descrição: O levantamento dos requisitos do produto, servirá para compor o *product backlog* inicial, ou seja, a visão do produto.

Parte Envolvida: *Product Owner.*

4.2.1.6 Planejamento do Projeto

Descrição: Em cima do que foi definido como a visão do produto é feito o planejamento do projeto, que contém o estudo de viabilidade, os requisitos, as iterações do projeto (módulos entregáveis para o cliente) com o objetivo de cada uma dessas e a formação do time.

Parte Envolvida: *Scrum Master e Product Owner*

4.2.1.7 Revisar Artefatos do Planejamento

Descrição: Essa atividade baseia-se na revisão de todos os artefatos criados até o momento, o *Product Owner* avalia e é o responsável.

Parte Envolvida: *Product Owner.*

4.2.1.8 Preparar ou Atualizar o ambiente do projeto

Descrição: Nesse momento o time (equipe, formado por analista de sistema, desenvolvedor, testador, DBA, etc) começa a preparar o ambiente do projeto.

Parte Envolvida: Time.

A fase subsequente do PPDSI_Ágil é a fase de execução, que tem a mesma estrutura desta seção.

4.2.2- Execução

Nesta fase o *product backlog* será evoluído, é definida a meta da *sprint*, bem como seu planejamento, o que irá ter ao final de cada *sprint*, quantas estórias vai entrar no *sprint*, após essa etapa é realizado o(s) *Sprint(s)*, ou seja, o ciclo de desenvolvimento, que neste caso é definido como um sub-processo, como mostra a Figura 17, no qual teremos as atividades relacionadas com a análise, projeto, construção, teste e integração do projeto, com estas sendo realizadas em paralelo e com as reuniões diárias, com o intuito de resolver os impedimentos o mais rápido possível, ao concluir esse sub-processo o time terá produzido o(s) incremento(s) entregável do produto, para assim constituir as versões intermediárias do projeto, não é obrigatório que em um *sprint* gere uma versão, esta pode ser constituída por vários sprints.

Ao final de cada *sprint* é realizado o *Sprint Review*, no qual o time se reúne com a participação do *product owner* e apresenta o que foi feito na *Sprint*, e, em seguida, se reúne para a *Sprint Retrospective* e verifica o que funcionou bem e o que pode melhorar nas próximas *Sprints*, em paralelo são feitos os testes adicionais, para garantir melhor qualidade no produto desenvolvido, e caso o *backlog* do produto esteja finalizado passa para próxima fase, senão volta para a atividade revisar e priorizar *backlog*. As reuniões *Sprint Review* e *Sprint Retrospective*, no caso de o cliente não poder participar, ocorrem apenas com a participação da equipe do projeto, do coordenador do projeto e da gerência da Fábrica.

Seguindo a mesma estrutura da fase anterior, as caixas em azul representam as atividades, os *gateways* em amarelo o fluxo de decisão, o círculo verde o início e o vermelho o fim da fase. A Figura 16 representa o fluxo de atividades desta fase.

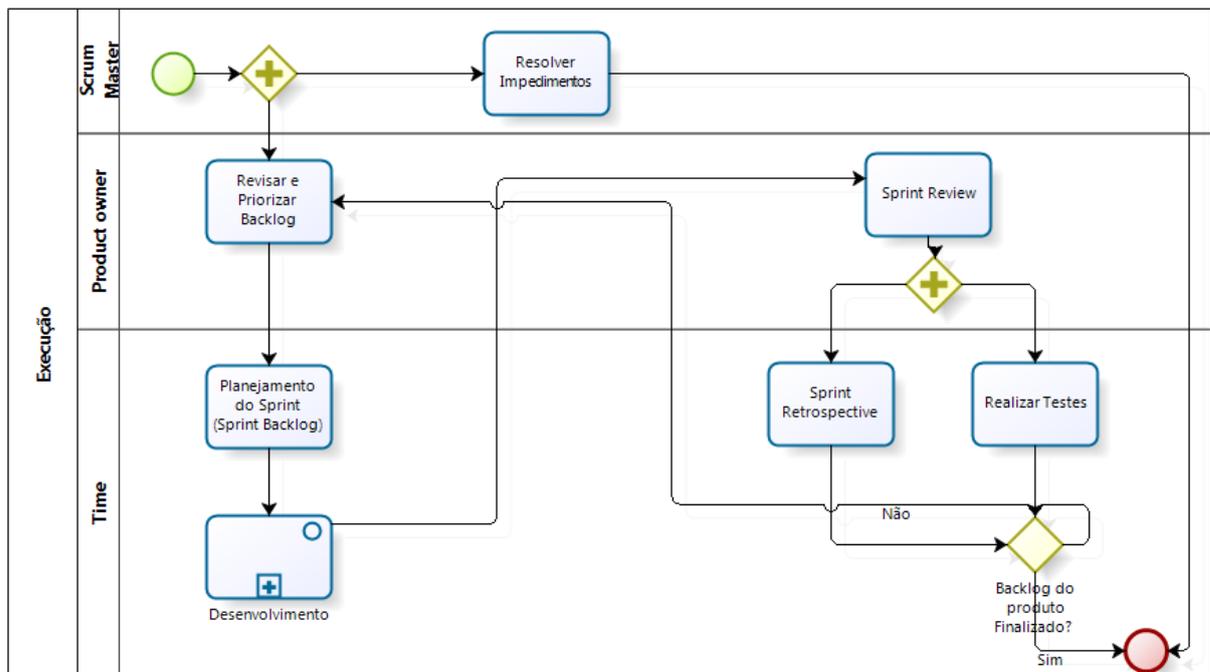


Figura 18 - Fluxo de Atividades da Fase de Execução

A seguir será detalhada cada atividade, com sua descrição e os papéis de quem realizam as mesmas.

4.2.2.1 Revisar e Priorizar *Backlog*

Descrição: Nesse momento o *product backlog* será revisado e conseqüentemente evoluído, para depois ser revisado e executado.

Parte Envolvida: *Product Owner*

4.2.2.2 Planejamento do *Sprint*

Descrição: É definida a meta da *sprint*, o que irá ter ao final de cada *sprint*, quantas histórias vão compor o *sprint*, a pontuação de cada item que compõe o *sprint* que é definido pelo time, bem como sua prioridade, após essa etapa é realizado o(s) *Sprint(s)*.

Parte Envolvida: Time, *Scrum Master* e *Product Owner*.

4.2.2.3 Desenvolvimento

Descrição: Esse é um sub-processo, no qual é realizado o(s) *sprint(s)*, gerando o(s) incremento(s) entregável do produto. Nessa etapa o *Scrum Master* apenas esta presente para resolver os impedimentos que possam ocorrer durante o *sprint*, com o objetivo de solucioná-los o mais rápido possível para não atrapalhar na produção da equipe. Todas as atividades desta etapa são realizadas em paralelo, e ao final é formada uma *tag*, contendo os itens elaborados.

Parte Envolvida: Time e *Scrum Master*.

4.2.2.4 *Sprint Review*

Descrição: É realizada a reunião, no qual o time se reúne com a participação do *product owner* e apresenta o que foi feito na *Sprint*, e o mesmo irá avaliar se o que foi feito está condizente com o esperado, caso contrário os itens que entraram em discordância e estiverem prioridade alta são resolvidos no próprio *sprint*, senão são colocados para os próximos *sprints*. O papel do *Scrum Master* é apenas de conduzir a reunião.

Parte Envolvida: Time, *Product Owner* e *Scrum Master*.

4.2.2.5 *Sprint Retrospective*

Descrição: É realizada uma reunião, no qual verifica o que funcionou bem e o que pode melhorar nas próximas *Sprints*, participa desta quem o time convida. O papel do *Scrum Master* é apenas de conduzir a reunião e a presença do *Product Owner* não é obrigatória.

Parte Envolvida: Time e *Scrum Master*.

4.2.2.6 Realizar Testes

Descrição: A tarefa de realizar testes é feito em paralelo com a *Sprint Retrospective* e são referente apenas aos testes adicionais (como teste de performance, stress, usabilidade, etc) para garantir melhor qualidade no produto desenvolvido.

Parte Envolvida: Time.

No sub-processo denominado desenvolvimento exibido na Figura 17, as atividades são realizadas em paralelas, com tarefas de análise, implementação, ajustes e testes, no qual estes são realizados os testes unitários, de integração e de sistema. Essas atividades são realizadas pelo time, cada integrante executa a atividade que lhe foi concebida.

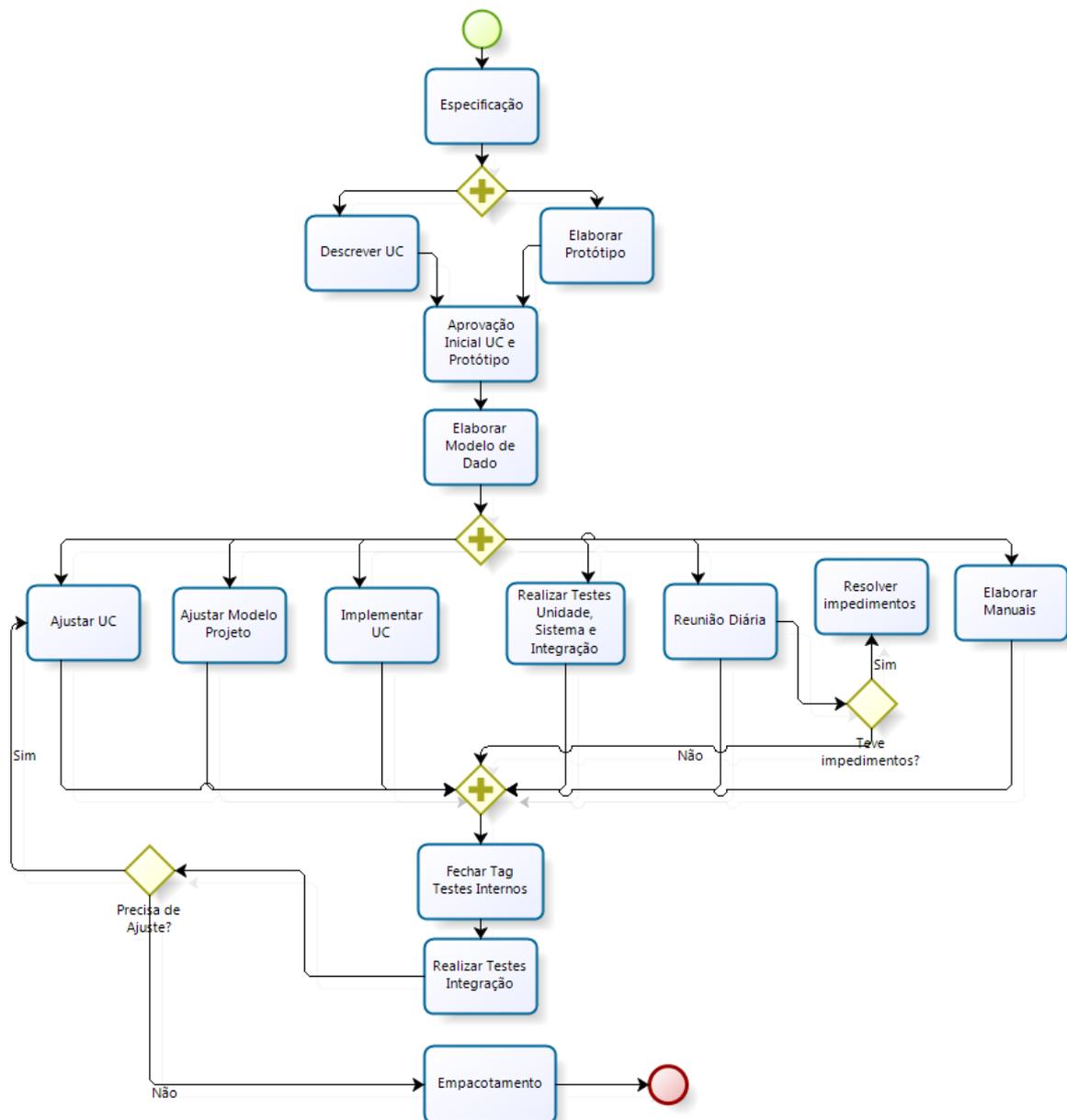


Figura 19 – Sub-processo Desenvolvimento

Como no início de todo desenvolvimento de software, as primeiras atividades a serem executadas são referentes à análise do sistema, no qual após a especificação da mesma, já podem ser elaborados em paralelo os casos de uso e os protótipos de tela, caso seja necessário, os artefatos gerados por essas atividades precisam ser aprovados, para depois elaborar o modelo de dados. As atividades seqüentes também são realizadas em paralelo e caso precise de algum ajuste nos casos de uso ou no modelo do projeto é realizado, depois é implementado os casos de uso seguido dos testes unitários.

Durante toda a realização dessas atividades acontecem as reuniões diárias com o time, no qual é dito por cada integrante o que fez no dia anterior, o que esta fazendo hoje e se tem algum impedimento que está dificultando prosseguir com o trabalho, caso exista, o *Scrum Master* é o responsável por resolvê-los. Dentro deste mesmo ciclo já podem ser desenvolvidos ou iniciados a documentação do sistema. Após a finalização das tarefas correspondentes é fechado uma *tag*, e o fluxo volta para a fase de execução.

4.2.3- Encerramento

Nesta fase é preparado o empacotamento ou distribuição do produto, o agrupamento da documentação, a formalização do projeto, a execução dos treinamentos para usuários finais sobre uso e operação do produto, pois seu planejamento foi realizado no desenvolvimento do projeto, e a aceitação do projeto ou fase, por fim, é realizada a finalização do mesmo documentando as lições aprendidas.

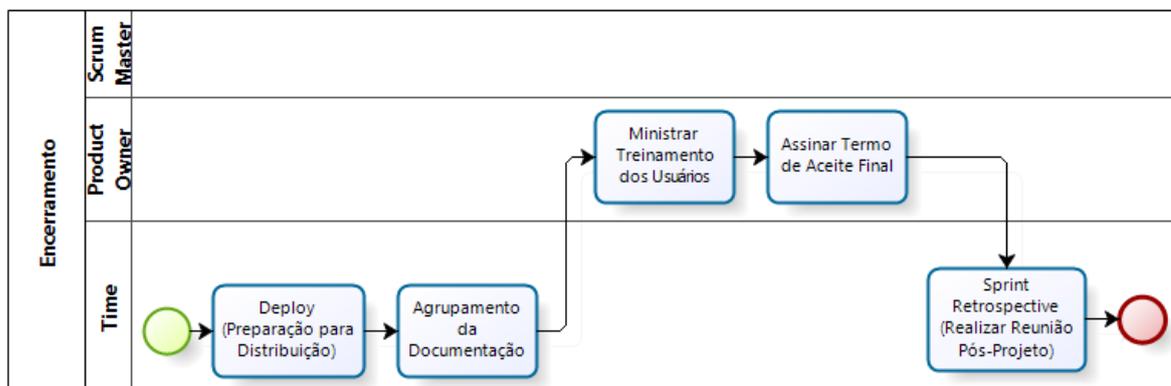


Figura 20 - Fluxo de Atividades da Fase de Encerramento

4.2.3.1 Preparação para Distribuição

Descrição: Preparação do ambiente para implantação. Trazer os dados originais da base do cliente e adaptá-lo a realidade da nova aplicação.

Parte Envolvida: Time.

4.2.3.2 Agrupamento da Documentação

Descrição: Será finalizada toda a documentação técnica do projeto que deverá ser entregue ao cliente, de acordo com os itens que foram estabelecidos pelo mesmo.

Parte Envolvida: Time.

4.2.3.3 Ministrando Treinamento dos Usuários

Descrição: O *Product Owner* deverá montar a estrutura do curso de treinamento, já que o mesmo detém das regras de negócio do cliente. Elaborando um conteúdo programático contendo os tópicos a ser ministrado.

Parte Envolvida: *Product Owner*.

4.2.3.4 Assinar Termo de Aceite Final

Descrição: O Cliente assina um termo aprovando o aceite final de que o objeto do contrato foi finalizado e entregue de acordo com suas perspectivas.

Parte Envolvida: *Product Owner*

4.2.3.5 Sprint Retrospective

Descrição: Ao término do projeto é realizada uma reunião com todos os membros envolvidos onde é feita uma retrospectiva de todo o processo realizado. É importante destacar quais os pontos mais positivos, negativos e as lições aprendidas para que os procedimentos estejam sempre em constante melhoria.

Parte Envolvida: Time, *Scrum Master* e *Product Owner*.

A próxima seção apresenta as métricas utilizadas na aplicação do processo, no qual as mesmas foram selecionadas utilizando o método GQM.

4.3- Métricas Utilizadas no Estudo de caso

As métricas foram definidas através do método GQM, apresentada na Seção 2.6.1, no qual definimos o objetivo do projeto e da avaliação das métricas, através deste objetivo, são levantados questionamentos, e as respostas destes são as metas que desejamos alcançar, ou seja, as métricas que serão avaliadas.

Objetivo

Avaliar o desempenho do processo da Infox, com o intuito de subsidiar e apoiar as informações gerenciais da organização, verificando o esforço das pessoas no processo, avaliando como atributo a quantidade de pessoas no projeto. Uma das atividades do processo é a verificação da gestão de projetos e dos requisitos usados com base no *framework Scrum*, possibilitando identificar o esforço e as atividades necessárias para atender as áreas de processo do nível G-MPS.Br. Verificando as entrega de funcionalidade com alto valor de negócio, a produtividade da equipe e a qualidade dos artefatos.

Questionamentos

- Qual o valor de negócio estimado das funcionalidades entregues?
- Qual o valor de negócio final das funcionalidades entregues?
- Qual a produtividade da equipe?
- Qual a porcentagem de estórias aceitas?
- Qual a porcentagem de erros nas histórias aceitas?
- Qual o número médio de solicitações de mudança sobre as histórias aceitas?
- Qual a quantidade de estórias rejeitadas pelo cliente?
- Quantas estórias foram acrescentadas durante a execução do projeto?

Métricas

- Valor de negócio da funcionalidade
- Número de estórias aceitas
- Número de Solicitações de Mudança por Funcionalidade num período
- Número de Horas utilizadas por funcionalidade

5- ESTUDO DE CASO DA APLICAÇÃO DO PPDSI_ÁGIL

O objetivo deste capítulo é apresentar o projeto selecionado para aplicação do processo, como também o experimento realizado com a finalidade de avaliar o processo concebido e em seguida os resultados obtidos. A organização do mesmo é dada da seguinte forma, a Seção 5.1 descreve o contexto do projeto utilizado para validar o processo, a Seção 5.2 relata a validação do processo concebido no projeto, a Seção 5.3 descreve a análise comparativa entre o PPDSI e o PPDSI_Ágil, e a Seção 5.4 apresenta as considerações finais deste trabalho.

5.1- Sistema Gerenciador de Serviços (SGS)

O projeto selecionado foi o SGS (Sistema Gerenciador de Serviços), que foi criado pela INFOX para controlar o fluxo de chamados permitindo o seu rastreamento, sendo cedido para utilização no Tribunal Regional Federal da 5ª Região (TRF-5ª), situado em Recife (PE), atendendo as exigências do contrato firmado entre as instituições.

O TRF-5ª solicitou melhorias no SGS para adequar ao contrato e necessidades internas de controle dos chamados abertos pela instituição e executados pela INFOX, desta forma, tornou-se necessário, a criação de um novo módulo, denominado Melhorias Gerais, no qual o objetivo era implementar as melhorias propostas pelo cliente, em atendimento aos requisitos especificados no contrato de prestação de serviços. Esse módulo foi desenvolvido utilizando o PPDSI, o processo desenvolvido pela INFOX, explicado no capítulo 3, e serviu para fazer a análise comparativa entre o PPDSI e o processo concebido.

Para aplicação do PPDSI_Ágil foi utilizado o mesmo projeto SGS, e foi desenvolvido o módulo de Relatórios Gerenciais de Serviços de Administração das Soluções em Produção. Neste módulo foram desenvolvidos 10 relatórios, dentre os quais temos: Relatório de Procedimentos Periódicos, Relatório de Monitoramento das Máquinas Servidoras, Relatório de Monitoramento das Soluções de Sistema, Relatório de Administração de Banco de Dados, Relatório de Administração de Dados, Relatório de Gestão de Configuração, Relatório de Gestão de Requisitos, Relatório de Gestão de Usabilidade, Relatório de Gestão de Qualidade e Relatório de Gestão de Arquitetura e Tecnologia. A seguir, será apresentado o resultado da aplicação do processo neste módulo.

5.2- Aplicação do Processo Concebido

Para aplicação do processo foi utilizado o módulo de Relatórios Gerenciais do SGS, como citado na seção anterior, esse foi alocado exclusivamente para ser aplicado utilizando o PPDSI_Ágil, no qual foi desenvolvido por uma equipe com quatro pessoas, e por se tratar de um projeto interno da empresa, foi possível a mudança de processo no trabalho de uma equipe em um projeto já em desenvolvimento.

Foram realizados dois *sprints* no desenvolvimento do módulo, ambos com duração de uma semana e produzindo os dez relatórios correspondentes. Para a alocação do tempo e esforço foi à própria equipe de desenvolvimento que estimou, avaliando a complexidade do relatório com base na experiência com projetos anteriores, no qual foi definido que um relatório de complexidade baixa teria o prazo de dois dias, média três dias e alta quatro dias. Antes de realizar o experimento, foi apresentado para equipe todo o processo, mostrando suas fases, e a seqüência de atividades.

Para tentar resolver ao máximo os impedimentos surgidos foram realizados todos os dias as reuniões diárias, no qual trouxe um valor considerável para equipe, foi possível resolver problemas que se não fosse à utilização dessa prática só iriam ser descobertos no final do prazo estimado para a tarefa.

Devido à pequena quantidade de pessoas no projeto, no qual tinha apenas uma analista, esta também foi alocada para realizar as tarefas de teste, o que foi bem significativo, e eliminou a ociosidade, quando as tarefas de desenvolvimento eram concluídas pelos desenvolvedores, a analista realizava os testes, e qualquer problema já reportava para o desenvolvedor correspondente.

O papel da autora deste trabalho foi realizar a coleta de métricas e ao mesmo tempo esclarecer dúvidas da equipe em relação ao *Scrum* já que o PPDSI_Ágil utiliza-o como base, portanto não interferiu em nenhum momento sobre a realização das tarefas pela equipe.

Na próxima seção, é apresentada a análise obtida com a comparação entre o processo PPDSI, atualmente utilizado pela Infox e o processo PPDSI_Ágil desenvolvido neste trabalho.

5.3- Análise Comparativa entre os Processos

Depois de apresentar o projeto no qual foi possível colocar em prática o PPDSI_ÁGIL, analisaremos comparativamente as abordagens de processos, tendo como foco o PPDSI e o PPDSI_ÁGIL.

Esses dois processos que são analisados têm pontos em comum que tornam a comparação possível: ambos possuem uma seqüência de atividades, realizadas por papéis, que geralmente são indivíduos ou a equipe, a fim de gerar artefatos ou produtos para serem entregues aos clientes. Os dois processos foram utilizados pela INFOX no desenvolvimento do SGS.

Para fazer a comparação entre os processos, utilizamos cada um em módulos distintos de um mesmo projeto, e depois transformando o que foi produzido em pontos de função, métrica explicada na Seção 2.7.

Como resultados encontrados nos projetos têm o módulo ‘Melhorias Gerais’, que utilizou como processo o PPDSI, e foram desenvolvidos quatro casos de uso, totalizando trinta e quatro pontos de função, e o módulo ‘Relatórios Gerenciais’ que utilizou o PPDSI_Ágil e foram desenvolvidos dez relatórios, totalizando quarenta pontos de função.

Os itens comuns para a análise comparativa dos dois processos foram selecionados a partir das métricas como apresentadas na Seção 4.3 e estão representados nas Tabelas 3 e 4, na qual a primeira apresenta os dados obtidos da execução do projeto Melhorias Gerais com o PPDSI, demonstrando para cada Caso de Uso do projeto a quantidade de pontos de função, o tempo de execução para desenvolver o mesmo, o número de requisitos associados ao caso de uso, que para tornar a comparação mais exata foram transformados em estórias, no qual cada caso de uso terá uma ou mais estórias, e uma estória está relacionada a uma tarefa(funcionalidade) que o usuário do sistema tem necessidade de executar, esses são aprovados pelo cliente e as solicitações de mudanças que foram registradas.

O tempo estimado pelo gerente do projeto que também é apresentado na Tabela 3 representa a quantidade de horas de análise e desenvolvimento de cada caso de uso, estas horas foram agrupadas para a Tabela 3 ficar análoga com a Tabela 4 e melhor entendimento na comparação de ambas. A maioria dos casos de uso está com o tempo estimado ultrapassando o tempo realizado, isso se deu devido não possuir a documentação atualizada do sistema, foi necessário ajustar juntamente com o fornecedor de requisitos, e também devido a um dos desenvolvedores ter sido realocado para este projeto momento antes de sua execução.

O número de solicitações de mudanças está relacionado aos itens que o cliente não aprovou, dentre os quais na sua maioria era referente a problemas de interface, ou seja, como os dados eram exibidos, no qual existia um padrão nas mudanças solicitadas.

Tabela 3 - Análise do projeto Melhorias Gerais

Caso de Uso	Ponto de Função	Tempo Estimado	Tempo Realizado	Estórias	Solicitações de Mudanças
Iniciar Chamado	8	19h	44h	6	4
Consultar Chamado	8	22h	44h	3	5
Painel do Usuário	10	44h	63h	3	1
Cadastro de Demandas	8	23h	53h	7	3
Total:	34	108hrs	204hrs	19	13

Neste exemplo, temos para o caso de uso Iniciar Chamado, um total de 6 estórias associadas ao mesmo, produzindo 8 pontos de função, em 44 horas, e totalizando 4 solicitações de mudanças requeridas pelo cliente.

A aplicação deste processo influenciou nos resultados, pois foram seguidas as suas fases, no qual ao fim de uma é que inicia a outra, o que pode ter ocasionado na quantidade de horas gastas ultrapassarem o tempo estimado.

Para o segundo projeto, que utilizou o PPDSI_Ágil, temos também para cada Caso de Uso a quantidade de pontos de função, o tempo de execução, a quantidade de estórias aceitas e as solicitações de mudanças.

Tabela 4 - Análise do módulo Relatórios Gerenciais

	Ponto de Função	Tempo Estimado	Tempo Realizado	Estórias Aceitas	Solicitações de Mudanças
Relatório de Procedimentos Periódicos	4	32h	24h	5/5	1
Relatório de Monitoramento das Máquinas Servidoras	4	32h	24h	½	1

Relatório de Monitoramento das Soluções de Sistema	4	32h	24h	2/3	1
Relatório de Administração de Banco de Dados	4	24h	36h	3/4	1
Relatório de Administração de Dados	4	24h	20h	2/3	-
Relatório de Gestão de Configuração	4	24h	32h	3/3	-
Relatório de Gestão de Requisitos	4	24h	20h	3/3	-
Relatório de Gestão de Usabilidade	4	24h	22h	3/3	-
Relatório de Gestão de Qualidade	4	24h	20h	3/3	-
Relatório de Gestão de Arquitetura e Tecnologia	4	24h	18h	3/3	-
Total:	40	264h	240h	28/32	4

Neste exemplo, temos para o caso de uso Relatório de Gestão de Configuração, um total de 3 estórias aceitas associadas ao mesmo, produzindo 4 pontos de função, em 32 horas, e nenhuma solicitação de mudança. No entanto, o tempo realizado ultrapassou o estimado devido ao desenvolvedor está executando 2 funções e alocado só meio período para desenvolver os casos de uso.

É apresentado na Tabela 4 o tempo estimado para cada relatório, sendo este definido pela equipe como citado na seção 5.2, temos dois relatórios que o tempo realizado ultrapassou o tempo estimado, isso ocorreu devido ao desenvolvedor que estava com esta tarefa, está alocado para desenvolver apenas meio período, ele assumia papel de AD (Administrador de Dados) pela manhã e pela tarde desenvolvia alguns relatórios juntamente com a equipe.

Os dados mostram também que muitos relatórios foram desenvolvidos com tempo menor do que o estimado, isso ocorreu devido à analista e o desenvolvedor estarem

trabalhando em conjunto, e qualquer dúvida era esclarecida no momento em que ele estava desenvolvendo o relatório, contudo facilitou bastante o trabalho de ambos e trouxe mais agilidade nas atividades executadas pela equipe.

A solicitação de mudança neste caso ocorreu no relatório de procedimentos periódicos, que acarretou mudança nos outros três relatórios, esta foi ocasionada por uma mudança no qual todos os procedimentos tinham que ser incluídos para calcular na fórmula definida e exibir todos os chamados por área, ou seja, uma mudança na regra de negócio.

Como consequência dos dados obtidos nas tabelas anteriores, temos a Tabela 5, que apresenta para cada projeto, o tempo médio e a produtividade alcançada. No qual a utilização do novo processo influenciou nos resultados, o número obtido para tempo médio foi resultado da fórmula abaixo:

$$\text{TempoMédio: } TC1 + TC2 + \dots + TCx / n$$

Onde,

TCx – Tempo de Execução do Caso de Uso identificado pelo número x

n – total de casos de uso

E a produtividade alcançada resulta da seguinte fórmula:

$$\text{Produtividade: } \text{quantidade de horas total do projeto} / \text{quantidade total de PF}$$

Onde,

PF – Pontos de Função

Tabela 5 - Tempo Médio e Produtividade dos projetos

	Melhorias Gerais – PPDSI	Relatórios Gerenciais – PPDSI_Agil
Tempo Médio	30hs	17hs
Produtividade	6 h/PF	4,4 h/PF

Os resultados apresentados na Tabela 5 demonstram que fazendo uso do processo PPDSI_ÁGIL são necessários 4h:24min (quatro horas e vinte quatro minutos) para executar um ponto de função, enquanto fazendo uso do processo PPDSI são necessários 6h (seis horas), resultando em um redução de 1h:24min com o uso do primeiro processo.

Portanto, em relação com o número total de estórias aceitas percebe-se que o PPDSI_Ágil entregou uma quantidade maior de estórias quando comparado ao PPDSI, e conseqüentemente possuiu mais estórias aceitas após a solicitação de mudança.

Como outra avaliação tem-se que o tempo estimado quando o mesmo é feito pela própria equipe, no caso do processo PPDSI_Ágil chega mais próximo ao total de horas de tempo realizado das tarefas como mostrado na Tabela 4.

A aplicação dos processos foi feita no mesmo projeto, mas com equipes distintas, no qual isto influenciou nos resultados encontrados, pois temos integrantes da equipe que não conhecia as regras de negócio do sistema, como também nunca tinha trabalho integrado à mesma.

Vale ressaltar que este resultado é apenas inicial e não pode ser generalizado, já que a validação do processo foi realizada num curto intervalo de tempo. Mas os relatos dão uma intuição que uma aplicação contínua do PPDSI_Ágil continuaria trazendo melhores resultados que o atual processo adotado na empresa.

Observa-se também que se houvesse a troca dos módulos na aplicação dos processos conseqüentemente o resultado não seria o mesmo, pois foi simples para a equipe definir o tempo estimado do módulo Relatórios Gerenciais, já que existe um padrão de como os relatórios são implementado, desta forma definir quanto tempo levaria para desenvolver foi relativamente fácil, e o outro projeto que foi desenvolver quatro casos de uso e sendo que alguns deste tem interligação com outros existentes e funcionando no sistema seria mais difícil definir a quantidade de horas previstas de cada.

Desta forma, como vantagem do novo processo tem-se que a técnica de determinar a quantidade de horas para o tempo estimado sendo realizado pela própria equipe traz bons resultados, pois traz um compromisso a mais para a mesma de cumprir o prazo que estimou, além de chegar o mais próximo do tempo gasto, pois ninguém melhor que eles para saber quanto tempo leva para desenvolver cada tarefa. Quando este método fica freqüente no trabalho da equipe, os prazos vão se tornando cada vez mais próximo do realizado, tendo como base resultados de projetos anteriores que foram estimados seguindo a mesma técnica.

Como outra vantagem percebe-se que a integração no trabalho em equipe é essencial em projetos de software, já que um artefato produzido por uma função serve como entrada de outra e cada trabalho gerado individualmente precisa ser integrados, então quanto mais perto os integrantes da equipe de um projeto estão mais as chances de obter sucesso no mesmo.

Ao final da aplicação do PPDSI_Ágil os membros da equipe foram questionados quanto aos benefícios do mesmo, e como resultado obteve os seguintes relatos:

“A metodologia com Scrum traz vantagens em relação à produtividade e melhora a comunicação entre a equipe”.

Desenvolvedor

“Interessante a ferramenta para todas as áreas (análise, desenvolvimento, testes), facilitando a integração das atividades entre si, não havendo brecha para ociosidade”.

Analista de Sistema

“Com a adaptação do Scrum ao PPDSI consegue atender de uma forma melhor o trabalho em equipe e contribui para melhores resultados”.

Desenvolvedor

Desta forma, como resultado final deste trabalho a próxima seção apresenta as considerações finais do mesmo.

5.4- Considerações Finais

Este capítulo apresentou os resultados da aplicação dos processos em um projeto de software escolhido, com o objetivo de avaliá-los buscando demonstrar benefícios do processo baseado no *Scrum*.

Com base na observação da aplicação dos processos, verificou-se que o PPDSI_Ágil apresentou um melhor desempenho em relação a produtividade dos artefatos gerados em comparação com o PPDSI. No entanto, devem-se levar em consideração os critérios analisados para a obtenção do resultado obtido por cada processo. O primeiro processo utiliza metodologia tradicional, já o segundo, utiliza práticas ágeis.

Levando em consideração apenas ao Processo Padrão de Desenvolvimento de Software Infox, que foi utilizado na comparação entre o processo concebido, percebemos que durante muitos anos conseguiu desenvolver com eficiência grandes projetos na empresa, firmando assim o nome da mesma no cenário nacional. Porém, com o passar dos anos, a Infox passou por um processo de expansão, investindo em capital humano e renovando suas tecnologias, adotando o que de mais atual existia no mercado. Mais que uma mudança, isso representou o início de uma nova cultura organizacional, na qual os antigos processos passaram a serem vistos como ultrapassados e/ou pouco eficientes. A ampliação do quadro de colaboradores acabou também tornando os processos existentes muito burocráticos.

No entanto, verifica-se que a institucionalização de um processo, deve ser sempre re-avaliada buscando a melhoria contínua, para atender as necessidades da organização, aumentando o nível de qualidade do produto e do processo, almejando satisfazer seus clientes. Desta forma, o processo é gerenciado e melhorado trazendo sempre benefícios para a mesma.

6- CONCLUSÃO

O exercício de definir e melhorar processos de software possibilita organizar e controlar os processos dentro do ciclo de vida do software. A melhoria de software consiste também na realização de ações que alteram seus processos, para que estes satisfaçam, de forma mais eficiente, os objetivos e necessidades das organizações, com a finalidade de reduzir custos, prazos e problemas relacionados a escopo, buscando aumentar sua produtividade, satisfação do cliente, o estabelecimento de uma linguagem comum e melhor comunicação entre toda a equipe, construindo uma visão compartilhada e ao mesmo tempo, uma estrutura para priorizar ações e projetos.

Desta forma tem-se que com a definição do novo processo, cujo objetivo era adoção do Scrum obtendo assim melhores níveis de desempenho, conseguiu resolver alguns problemas existentes, tais como, melhor comunicação entre a equipe, definição de prazos mais reais e resolução dos impedimentos o mais rápido possível, com o objetivo de não atrapalhar no trabalho da equipe e assim não comprometer o prazo estabelecido.

Nesse contexto, o processo concebido e apresentado nesse trabalho propõe a melhoria do processo da Infox, englobando práticas do framework Scrum. Com o objetivo de avaliá-lo o mesmo foi comparado com o processo atual da organização, e foram analisados os seus resultados. Dentre os quais se destacam a integração da equipe, formando um time apto para desenvolver suas tarefas no tempo estabelecido pela mesma, com isto também resolver os impedimentos quando estes estiverem relacionados com a dificuldade que algum dos integrantes esta naquele momento e tentar resolvê-los.

A seguir, são apresentadas as principais contribuições deste trabalho, como também alguns trabalhos relacionados com resumos de seus estudos de caso de natureza semelhante a este projeto.

6.1- Principais Contribuições

As principais contribuições desse trabalho são:

- Definição de um Processo Padrão de Desenvolvimento de Software baseado no Scrum;
- Análise do processo concebido em um projeto de desenvolvimento de software real.

- Análise das necessidades do MPS.br nível G para implantação do Framework Scrum;

6.2- Trabalhos Relacionados

6.2.1- PYXIS Tecnologia

O trabalho realizado pela Pyxis Tecnologia, empresa localizada na cidade de Aracaju, Sergipe, descreve a iniciativa em conceber um processo de gerenciamento de projetos de desenvolvimentos de software baseado em metodologias ágeis, ferramentas e frameworks sedimentados no mercado, em busca da certificação nível G do modelo de qualidade MPS.BR(NETO PEDRO, 2010). A empresa é apresentada como de pequeno porte, criada no ano de 2003 e atua no segmento de desenvolvimento de software, terceirização e serviços especializados.

Na execução do trabalho foi apresentado o modelo MPS.BR e as exigências do nível G, como também foram abordados conceitos referentes à gestão de projetos segundo o PMI. A partir desta descrição, foi definida uma estratégia para implementação do projeto. A estratégia foi dividida da seguinte maneira: Estudo realizado dos assuntos relacionados; Definição de uma estratégia para alinhamento das práticas das metodologias estudadas com as exigências do modelo MPS.br; Avaliação dos recursos existentes na empresa, incluindo ferramentas, pessoas e infra-estrutura; Definição do processo customizado para o ambiente VSTS e Validação do processo quanto à certificação no nível G do MPS.BR.

O resultado concebeu um novo processo para a empresa, com fluxos, papéis, artefatos de entrada e saída, e permitiu à empresa uma auto-avaliação que contou com a participação de toda a equipe técnica. Esta avaliação determinou o estágio da empresa em relação aos requisitos do modelo como também despertou nos colaboradores envolvidos o comprometimento na melhoria dos pontos fracos detectados.

Com o modelo definido novos documentos foram criados, modificações foram realizadas e compromissos foram assumidos pelos membros da equipe. Não foi utilizado de uma experimentação prática para validação ou ajuste do processo proposto, visto que o foco era validar um processo ágil, que atendesse aos resultados esperados do MPS.br para a conquista da certificação nível G do modelo MPS.br.

6.2.2- RIGHTWAY Consultoria e Sistemas

Este outro trabalho teve como objetivo apresentar um diagnóstico da implementação do nível F do MR-MPS.br com práticas da metodologia ágil Scrum, através do estudo de caso da empresa Rightway Consultoria e Sistemas (Catunda Edmar, Nascimento Camila, Cerdeiral Cristina, Santos Gleison, Rocha Ana Regina, 2010).

A Rightway esta sediada no Rio de Janeiro, criada em 1995, especializada em desenvolvimento de soluções e projetos de Tecnologia da Informação que atua em quatro linhas de serviços: sistemas sob medida, infraestrutura, *business intelligence* e alocação de pessoal, é especializada em desenvolvimento de soluções e projetos de Tecnologia da Informação para empresas de médio e grande portes em diversos segmentos do mercado.

A empresa desejava implementar processos para organizar melhor o desenvolvimento de software, permitindo um maior controle e alcançando uma melhor qualidade nos seus produtos; porém, sem perder a agilidade nos seus projetos.

Na execução deste trabalho o segmento definido foi: Análise das práticas, processos e procedimentos da empresa; Primeira Versão do Processo Padrão, buscando manter as práticas ágeis já utilizadas e atender às necessidades da empresa; Projeto Piloto: O processo padrão foi utilizado em um primeiro projeto com o apoio de ferramentas; Capacitação *Scrum Master*: a gerência da Fábrica e todos os coordenadores de projetos da empresa fizeram o curso e se tornaram *Certified Scrum Masters*, aumentando o conhecimento; Segunda Versão do Processo Padrão foi elaborada mais voltada para as práticas do Scrum e mais adaptada à realidade da empresa; Utilização do processo padrão nos demais projetos da empresa, com o apoio de ferramentas que tornassem mais ágil a execução do processo e automatizassem algumas atividades.

Este trabalho apresentou a metodologia utilizada para implantar as melhorias de processos necessárias para alcançar o nível F do modelo MR-MPS na empresa. Como resultado disseminou a combinação de práticas ágeis e modelos de maturidade e capacidade em processos de desenvolvimento, e na seleção de ferramentas de apoio para a execução dos processos. Com o modelo definido a empresa determinou os processos organizacionais e está executando nos seus projetos, estando em estágio avançado no projeto de implementação das melhorias de processo necessárias para alcançar o nível F, aguardando a avaliação da Softex no tempo previsto.

7- REFERÊNCIAS BIBLIOGRÁFICAS

SOFTEX. **Sociedade Brasileira para Promoção da Exportação de Software - Histórico**. 2009. <Disponível em http://www.softex.br/_asoftex/historico.asp#>. Acessado em abril de 2011.

_____. **MPS.BR Melhoria de Processo de Software Brasileiro - Guia Geral**. 2009a. Disponível em <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2009.pdf>. Acessado em outubro de 2009.

_____. **MPS.BR Melhoria de Processo de Software Brasileiro – Site do Programa**. 2009b. Disponível em <http://www.softex.br/mpsbr/_home/default.asp>. Acessado em abril de 2011.

AGILE. **Agile Manifesto**. 2001. Disponível em <<http://agilemanifesto.org/>>. Acessado em abril de 2011.

ALLIANCE. **Scrum Alliance**. 2009. Disponível em <<http://www.scrumalliance.org/>>. Acessado em abril de 2011.

INFOX. Infox Tecnologia da Informação Ltda. Disponível em <www.infox.com.br>. Acessado em abril de 2011.

PEDRO SILVA NETO, 2010. Estudo de Caso de um Processo de Gerenciamento de Projetos de Software baseado em Métodos Ágeis e no Modelo MPS.br.

Catunda Edmar, Nascimento Camila, Cerdeiral Cristina, Santos Gleison, Rocha Ana Regina, 2010, **Implementando o Nível F do MR-MPS com Práticas da Metodologia Ágil Scrum**.

SCHWABER, Ken. **Scrum Development Process**. Burlington, Massachussets, 1995. Disponível em <<http://jeffsutherland.com/oopsla/schwapub.pdf>>. Acessado em abril de 2011.

KOSCIANSKI, André; SOARES, Michel dos Santos, **Qualidade de Software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. 2. Ed. –São Paulo: Novatec Editora, 2007.

BIZAGI, Process Modeler - BPM Suite . Disponível em <<http://www.bizagi.com/>>. Acessado em agosto de 2011.

KNIBERG Henrik, **Scrum e XP Direto das Trincheiras. Como nós fazemos Scrum**. InfoQ, 2007.

ABNT – Associação Brasileira de Normas Técnicas. Disponível em <<http://www.abnt.org.br>>. Acessado em 20 de setembro de 2011.

PRESSMAN, Roger S. **Engenharia de Software**. 6ª Edição. McGraw-Hill Editora, 2006.