

**Universidade Federal de Santa Catarina
Departamento de Informática e estatística
Sistemas de Informações**

**PROCESSO UNIFICADO PARA GESTÃO DE LIÇÕES
APRENDIDAS**

Soraia Barbosa de Almeida

**Florianópolis - SC
2012/1**

Soraia Barbosa de Almeida

PROCESSO UNIFICADO PARA GESTÃO DE LIÇÕES APRENDIDAS

Monografia de Conclusão de Curso apresentada ao Curso de Sistemas de Informação da Universidade Federal de Santa Catarina, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Florianópolis - SC

2012/1

Soraia Barbosa de Almeida

PROCESSO UNIFICADO PARA GESTÃO DE LIÇÕES APRENDIDAS

Monografia de Conclusão de Curso apresentada ao Curso de Sistemas de Informação da Universidade Federal de Santa Catarina, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Florianópolis, SC, 15 de maio de 2012.

Prof. João Cândido Dovicchi
(Responsável e Orientador)

Prof. Vítório Bruno Mazzola
(Banca Examinadora)

Prof. José Eduardo De Lucca
(Banca Examinadora)

*"A maioria das pessoas pensa no sucesso e no fracasso como opostos,
mas eles são ambos produtos do mesmo processo."*

(Roger von Oech)

Para Norberto e Lourdes, meus pais,
pela oportunidade da vida e
por me abençoarem no céu e na terra.

AGRADECIMENTOS

Agradeço aos professores e a universidade por toda a dedicação e estrutura necessária para meu desenvolvimento pessoal e profissional. Especialmente ao professor Doutor em Engenharia e Mestre em Artes João Cândido Dovicchi, o qual antes de qualquer titulação é um ser humano maravilhoso, agradeço pelo voto de confiança, respeito e amizade.

Agradecimento especial a minha família, Lourdes Barbosa, Nuria, Renata e Júlia pelo apoio incondicional nas decisões que eu tomei, pela compreensão no meu isolamento durante a concentração e dedicação aos estudos e trabalho.

Aos colegas da empresa Nexxera e as pessoas que eu convivi nesses últimos cinco anos e que contribuíram de alguma forma com o resultado deste trabalho, alguns nomes: Satyro Oliveira, Marcia Back, Antônio Marcos Da Ré e Denise Ebone.

Agradecimento especial a Ricardo Debiazi Zomer, namorado, companheiro e melhor amigo pelos inúmeros gestos de amor, por se esforçar em tentar entender o que eu estava estudando, por me aguardar com sopa ou chocolate quente ao final das aulas em noites frias, pelas inúmeras correções que fez deste trabalho, por todo incentivo, apoio e críticas fundamentais para a minha graduação.

Por fim, a Deus por tudo.

RESUMO

Este trabalho aborda questões relacionadas ao processo de gerenciamento de lições aprendidas em projetos de software. Foi concebido um modelo de processo que permitirá que o conhecimento seja registrado, armazenado e disseminado em uma organização. O modelo proposto teve como objetivo ser compatível com os principais frameworks de melhores práticas de controle e gerência de TI. Foram eleitos para o desenvolvimento deste trabalho um framework para gestão (PMI), qualidade (CMMI), desenvolvimento (RUP) e desenvolvimento ágil (SCRUM). Foi realizada uma comparação entre os modelos com foco na gestão de lições aprendidas e, em seguida, concebido um processo único para as atividades de cada framework. Para validar o processo proposto foi realizado um estudo de caso em uma organização de software. O processo criado mostrou-se flexível e aderente aos modelos estudados neste trabalho.

Palavras chaves: Lições Aprendidas, Gerenciamento de Projetos, Metodologia Ágil, PMI, CMMI, RUP e SCRUM.

ABSTRACT

This work consisted in adress issues related to the process of management lessons learned in software projects. This will allow knowledge to be registered, stored and disseminated in a software organization. The model was proposed in order to be compatible with the main frameworks of best practices for TI control and management. For the development of this work, were elected a framework for management (PMI), quality (CMMI), development (RUP) and agile development (SCRUM). A comparison was made between models with a focus on management lessons seized, then designed a process unifying the activities of each framework. To validate the proposed process, a study was conducted in a software organization. The process created is flexible and showed adherent to the models studied.

Keywords: Lessons Learned, Project Management, Agile Methodology, PMI, CMMI, RUP and SCRUM.

LISTA DE FIGURAS

Figura 1 – Gráfico do caos em projetos de software	17
Figura 2 - Aspectos considerados na Metodologia de GP.....	18
Figura 3 - Áreas de Conhecimento de Gerenciamento de Projetos.....	24
Figura 4 - Conexão entre os Grupos de Processos.....	25
Figura 5 - Processo de Encerramento de Projeto.....	27
Figura 6 - Análise de <i>Pós-projeto</i>	29
Figura 7 - Etapas da realização de PostmemJiro	30
Figura 8 – Método KJ.....	31
Figura 9 - Diagrama de Ishikawa	31
Figura 11 - Adaptação - Funcionamento de um Experience Factory.....	33
Figura 12 – Frameworks e sua data de lançamento.....	34
Figura 13 - Representação por estágios.....	37
Figura 14 - Representação contínua	37
Figura 15 - Ciclo da metodologia scrum – sprint.....	43
Figura 16 - Gráfico das iterações do RUP	45
Figura 17 - Um papel, suas atividades e artefatos.....	48
Figura 18 - Fases do Processo Proposto.....	58
Figura 19 – Fase de Coleta.	59
Figura 20 – Fase de Processamento.....	61
Figura 21 – Fase de Armazenamento	63
Figura 22 – Fase de distribuição.....	64
Figura 23 - Fluxo do processo proposto	65
Figura 24 – Diagrama de Causa e Efeito.....	73

LISTA DE TABELAS

Tabela 1 - Análise de Lições aprendidas no PMI	50
Tabela 2 - Análise de Lições aprendidas no CMMI	52
Tabela 3 - Análise de Lições aprendidas no RUP	53
Tabela 4 - Análise de Lições aprendidas no SCRUM	54
Tabela 5 – Comparativo dos modelos	55
Tabela 6 – Modelo de Descrição das Características	56
Tabela 7 – Fase de Coleta	59
Tabela 8 - Fase de Processamento	61
Tabela 9 – Fase de Armazenamento	63
Tabela 10 –Fase de Distribuição.....	64
Tabela 11 - Papel/Responsabilidade dos Membros da equipe	69
Tabela 12 – Resultado do Questionário	70
Tabela 13 - Tabela de ocorrências discutidas	72
Tabela 14 – Parte da documentação	75
Tabela 15 - Tabela de Avaliação do Processo	76
Tabela 16 – Avaliação do Processo	77

LISTA DE ABREVIATURAS E SIGLAS

PMBOK - *Project Management Body of Knowledge*

PMI - *Project Management Institute*

SIGP - Sistema de Informações do Gerenciamento de Projetos

UFSC - Universidade Federal de Santa Catarina

GP - Gerenciamento de projetos

SGP- Software para gerenciamento de projetos

CMMI- *Capability Maturity Model® Integration*

CMMI - DEV - *CMMI for Development*

GC - Gestão do Conhecimento

TI – Tecnologia da Informação

IEC - International Electrotechnical Commission

ISO - International Organization for Standardization

IT –Information Technology

UML - Unified Modeling Language

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO	15
1.1 Considerações Iniciais	15
1.2 Contextualização.....	15
1.3 Problema.....	18
1.4 Objetivo Geral	19
1.5 Objetivos Específicos.....	20
1.6 Metodologia.....	20
1.7 Organização do Trabalho.....	21
CAPÍTULO 2 FUNDAMENTAÇÃO TEÓRICA	22
2.1 Gerenciamento de Projetos.....	22
2.2 Encerramento de Projeto	26
2.3 Técnicas de Lições Aprendidas	27
2.3.1 Análise de Pós-Projeto	29
2.3.2 Oficina de Experiências	32
2.4 Modelos de Referência	33
2.4.1 PMI	34
2.4.2 CMMI	35
2.4.3 Projetos de Software Agile.....	39
2.4.4 SCRUM.....	40
2.4.5 RUP	44
2.4.6 Papéis e Atividade de uma Organização de Software	48
CAPÍTULO 3 ESTADO DA ARTE.....	49
3.1 Análise do Estado da Arte.....	49
3.1.1 Lições Aprendidas no PMI	50
3.1.2 Lições Aprendidas no CMMI-DEV	51

3.1.3	Lições Aprendidas no RUP	52
3.1.4	Lições Aprendidas no SCRUM.....	53
3.2	Definição do Processo Genérico	56
3.3	Fases do Processo	57
3.3.1	Coleta.....	58
3.3.2	Processamento	61
3.3.3	Armazenagem.....	63
3.3.4	Distribuição.....	64
CAPÍTULO 4 ESTUDO DE CASO		66
4.1	Metodologia do Estudo de Caso.....	66
4.2	Organização	67
4.3	Projeto Alvo	67
4.4	Relatório do Estudo de Caso	69
4.5	Avaliação e Análise dos Dados	76
CAPÍTULO 5 CONCLUSÕES.....		78
5.1	Considerações Finais	78
5.2	Sugestão de Trabalhos Futuros.....	80
REFERÊNCIAS.....		81
APÊNDICES		84
APÊNDICE A - Questionário de lições aprendidas		84
APÊNDICE B - Resultado do Qualitativo do Questionário.....		89

CAPÍTULO 1

INTRODUÇÃO

1.1 Considerações Iniciais

Este documento apresenta o trabalho desenvolvido como atividade da disciplina INE5632 – Projetos II. O projeto realizado é relacionado com as áreas de gerenciamento de projetos, engenharia de software e processo.

Nesta seção são apresentados: contextualização, problema, objetivos e a metodologia no desenvolvimento deste trabalho.

1.2 Contextualização

O desafio hoje, em tempos de mudanças rápidas, está em gerenciar atividades nunca antes realizadas e que podem jamais se repetir no futuro. Assim, o gerenciamento de projetos pode ser definido como o planejamento, a programação e o controle de uma série de tarefas integradas à forma de atingir seus objetivos com êxito (KERZNER, 2007).

Nos últimos anos, o gerenciamento de projetos tem crescido de forma acentuada no mundo. As associações do PMI (Project Management Institute), conhecidas como capítulos, contam com 11.330 membros associados no Brasil, a cidade de São Paulo conta com 3.142 membros é o 12º maior capítulo no mundo em quantidade de membros.

No relatório de *Benchmarking* em GP do Brasil, de 2010, houve a participação de 460 organizações, distribuídas em 18 setores da economia, sendo o setor de TI - Tecnologia da Informação - o mais citado, com 18% das organizações participantes. Entre os produtos/serviços produzidos pelo setor de TI no Brasil, destaca-se a

consultoria de sistemas e o desenvolvimento de software com 43% da receita bruta (IBGE, 2011).

Geralmente sistemas de softwares são concebidos através de projetos. Os projetos de software são mais difíceis de serem gerenciados. O SEI - Software Engineering Institute, que em 1993, constatou que o principal problema que aflige as organizações de software é gerencial: “as organizações precisam vencer o seu buraco negro que é o seu estilo de gerenciar de maneira informal, sem métodos e sem técnicas”. Os fracassos no desenvolvimento de projetos de software são amplamente conhecidos e divulgados, como por exemplo, o relatório do *The Chaos Report* realizado pelo *The Standish Group*. Tal relatório foi encomendado pelo Departamento de Defesa Americano, e apontava, em 1994, que apenas 16% dos projetos de software obtinham sucesso, ou seja, eram concluídos no tempo, orçamento e qualidade de acordo com o planejado. Em 2011 o *Standish Group*, apresentou em seu relatório *CHAOS Report 2010* o seguinte resultado referente ao caos em projetos de software:

- 37% obtiveram sucesso (no prazo, dentro do orçamento e com escopo completo);
- 42% mudaram (atrasaram, estourou o orçamento, e/ou reduziram escopo);
- 21% falharam (cancelados ou nunca usados).

Houve um notável aumento no número de projetos de software com sucesso nos últimos 16 anos. Entre os motivos responsáveis por essa melhoria, destacam-se:

- Em 1994 foi criado e difundido o modelo de maturidade de desenvolvimento de software, o CMM - *Capability Maturity Model*.

Atualmente, o modelo de referência é o CMMI - *Capability Maturity Model Integration* na sua versão 1.3 (SEI, 2011).

- Em 1996 houve a criação e difusão do PMBOK - *Project Management Body of Knowledge* -, o qual reúne conhecimentos em gerenciamento de projetos. O guia PMBOK conta com novas edições lançadas em 2000, 2004 e 2008.

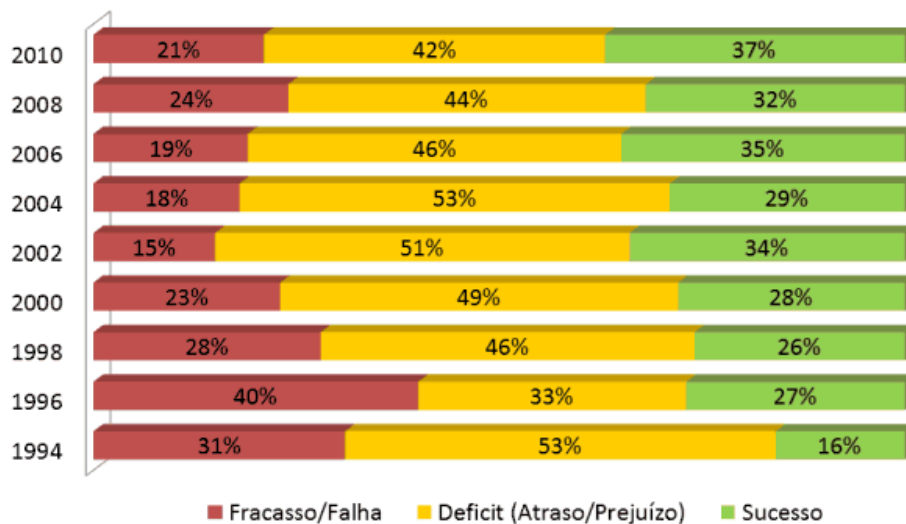


Figura 1 – Gráfico do caos em projetos de software
Fonte: *Standish Group, Chaos Manifesto 2001*

Para contribuir com o aprendizado em projetos, há necessidade de compartilhar conhecimento de forma a melhorar a relação entre os envolvidos, isso pode ocorrer através do gerenciamento de lições aprendidas. A documentação sistemática de percalços, erros ou armadilhas potenciais ajuda a reduzir os riscos do projeto (FERENHOF, 2011). A partir de uma perspectiva de longo prazo, um projeto de aprendizagem sistemática permite a empresa desenvolver competências de projeto que levam a uma vantagem competitiva sustentável (SCHINDLER, 2003).

Organizações podem economizar custos consideráveis, que resultam do trabalho redundante e a repetição de erros. O presente trabalho visa a abordar

questões relacionadas ao processo de lições aprendidas em projetos. Propõe-se a apresentar um processo para coleta, análise e disseminação de lições aprendidas em uma organização, que seja de compatível com os frameworks de desenvolvimento de software em evidência.

1.3 Problema

No Estudo de Benchmarking de GP 2010, foi constatado que 67,9% das empresas consideram a integração em seu gerenciamento e que apenas 37% consideram as aquisições. Sendo o grupo de processos de encerramento parte das áreas de integração e aquisições, pode-se inferir que, poucas são as empresas de TI que o praticam no Brasil, quando comparado com outros grupos de processos.

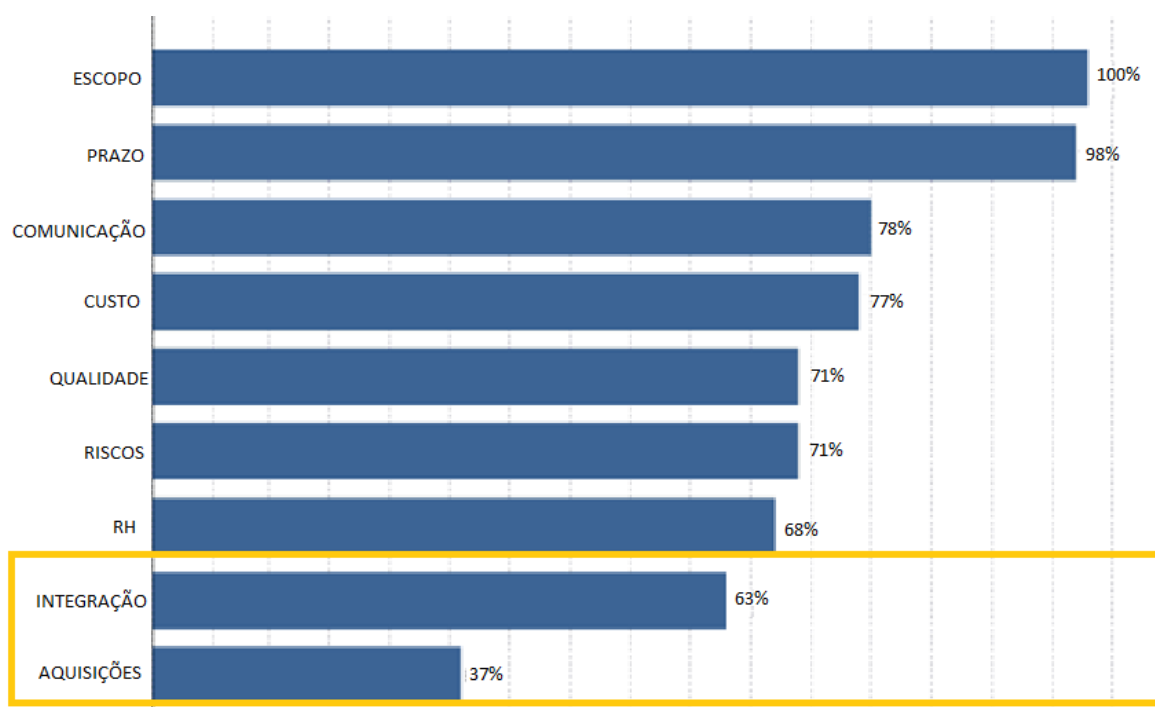


Figura 2 - Aspectos considerados na Metodologia de GP.

Fonte: Estudo de Benchmarking em Gerenciamento de Projetos Brasil 2010, Project Management Institute – Chapters Brasileiros.

A não implementação do grupo de processos de encerramento e na coleta de lições aprendidas de projetos pode ocasionar uma série de problemas, como por

exemplo:

- Repetição, em novos projetos, de desvios e problemas ocorridos anteriormente, devido a não realização de retrospectivas de projeto;
- Falta de percepção do desempenho final do projeto, visto que a declaração de sucesso ou falha de um projeto deve ocorrer no encerramento, levando-se em consideração o cumprimento do custo, prazo e escopo planejados.

Em todos os projetos de software, os integrantes ganham conhecimento e experiências que podem beneficiar futuros projetos, bem como o desenvolvimento profissional de cada membro participante (BIRK, 2002). Infelizmente, muito deste conhecimento permanece despercebido e muitas vezes não são compartilhados entre indivíduos ou equipes (BIRK, 2002). Apesar da importância do encerramento, problemas com restrições orçamentárias, marco oportuno para realização, a escassez de recursos ou mesmo a falta de interesse da equipe, todos entram em coexistência durante fase de encerramento (Sarfraz, 2009). O fim de um projeto é, portanto, o fim da aprendizagem coletiva. O pessoal envolvido passar para novos projetos ou são reintegrados nas suas funções de linha, amnésia organizacional começa (SCHINDLER, 2003).

Este trabalho pretende responder a pergunta: é possível implementar um processo unificado para o gerenciamento de lições alinhado aos principais frameworks de melhores práticas de controle e gerência de TI?

1.4 Objetivo Geral

Este trabalho possui como objetivo geral, a criação de um processo genérico para o gerenciamento de lições aprendidas aplicado em projetos, que permitirá que

o conhecimento seja registrado, armazenado e disseminado em uma organização de software. Tal modelo pretende ser compatível com os principais frameworks de melhores práticas de controle e gerência de TI.

1.5 Objetivos Específicos

Para que o objetivo geral seja alcançado, faz-se necessário que os objetivos específicos descritos abaixo sejam alcançados:

- OE 01. Contextualizar lições aprendidas dentro do contexto de desenvolvimento de software.
- OE 02. Modelar um processo genérico para o processo de encerramento alinhado com os principais frameworks de melhores práticas de controle e gerência de TI.
- OE 03. Aplicar o processo modelado em uma organização de desenvolvimento de software.
- OE 04. Avaliar o processo modelado através de um estudo de caso.

1.6 Metodologia

De acordo com os objetivos da pesquisa, denota-se uma pesquisa exploratória. Estas pesquisas têm como objetivo proporcionar maior familiaridade com o problema com vistas a torná-lo mais explícito (GIL, 2002).

Etapas da Pesquisa

A pesquisa do presente trabalho será realizada em cinco etapas:

Etapa 1 - Fundamentação teórica: análise da fundamentação teórica sobre o gerenciamento de lições aprendidas em projetos de software, com base em análise de literatura. A fundamentação teórica será desenvolvida basicamente com pesquisa

bibliográfica a partir de artigos, periódicos e livros, classificando-se como bibliográfica a pesquisa desenvolvida a partir de material já elaborado (GIL, 2002).

Etapa 2 - Análise do estado da arte: a partir da fundamentação teórica será modelado um processo genérico para o gerenciamento de lições aprendidas.

Etapa 3 - Definição de um processo genérico para o gerenciamento de lições aprendidas e a modelagem dos artefatos envolvidos.

Etapa 4 - Estudo de caso, será planejado e executado um estudo de caso, aplicando em uma organização de software o processo concebido na etapa anterior.

Etapa 5 - Avaliação e Conclusões, será realizada a avaliação do processo implementado e realizadas as conclusões.

1.7 Organização do Trabalho

Este trabalho está organizado da seguinte maneira: no *Capítulo 1* são apresentadas as questões que levaram à realização deste trabalho, assim como os objetivos gerais e específicos. O *Capítulo 2* trata da fundamentação teórica. São apresentadas e analisadas as referências dentro do escopo deste trabalho, abordados os termos de engenharia de software, gerenciamento de projetos e processo, assim como técnicas para execução do gerenciamento de lições aprendidas. O *Capítulo 3* apresenta a definição de um processo genérico para o gerenciamento de lições aprendidas e a modelagem dos artefatos envolvidos. No *Capítulo 4*, é realizada a implementação do processo de gerenciamento de lições aprendidas, etapa na qual é realizado um estudo de caso em uma organização de software. Por fim, o *Capítulo 5* apresenta a avaliação do processo implementado, bem como as conclusões deste trabalho e apontamentos para trabalhos futuros.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Para definir os processos que tangem o conceito de lições aprendidas, é necessário que outros conceitos fortemente ligados ao escopo desse trabalho sejam bem compreendidos. A fundamentação inicia-se com a apresentação dos termos ligados a projetos e gerenciamento de projetos com enfoque no grupo de processos de encerramento e, é seguida, por uma análise de técnicas e ferramentas de gerenciamento de lições aprendidas.

2.1 Gerenciamento de Projetos

Um projeto consiste num esforço temporário empreendido com objetivo pré-estabelecido, definido e claro, e possui as seguintes características: cada projeto cria um produto, serviço ou resultado exclusivo; são realizados por pessoas; são empreendimentos não repetitivos; possuem início, meio e fim bem definidos; duração e recursos limitados, numa sequência de atividades relacionadas (PMBOK, 2009). O CMMI-DEV 1.3 trata o conceito de projeto de forma mais específica definindo como um conjunto de recursos interligados e gerenciados, o qual fornece um ou mais produtos para o consumidor ou usuário.

Quando não há uma definição de início e fim e quando o empreendimento é repetitivo e o esforço é permanente, as atividades podem ser classificadas como operações, as quais são tipicamente realizadas de maneira contínua. As operações oferecem suporte para os projetos da organização, mas não estão ligadas diretamente aos processos de gerenciamento de um projeto, como também não geram um produto (PMI, 2008).

As nove áreas de conhecimento que caracterizam os principais aspectos envolvidos em um projeto e no seu gerenciamento são:

- **Gerenciamento de Integração do Projeto** – Envolve os processos necessários para garantir que os vários elementos de um projeto sejam propriamente coordenados.
- **Gerenciamento do escopo do projeto** - Envolve os processos necessários para assegurar que o projeto tenha definido todo o trabalho a ser realizado até o encerramento, e tão somente o trabalho necessário, para complementar de forma bem sucedida o projeto.
- **Gerenciamento de tempo do projeto** - Envolve os processos para garantir o encerramento do projeto no tempo certo. Consiste da definição, ordenação e estimativa de duração das atividades e de elaboração e controle de cronogramas.
- **Gerenciamento de custos do projeto** – Nessa área são tratados os processos relacionados com o orçamento, visa garantir o encerramento do projeto dentro do orçamento aprovado. Consistem no planejamento dos recursos, estimativa, orçamento e controle de custos.
- **Gerenciamento da qualidade do projeto** - Envolve os processos para assegurar que o projeto satisfaça as necessidades para o qual foi criado. Nesta área são controladas as políticas e procedimentos de qualidade.
- **Gerenciamento de recursos humanos do projeto** – Envolve os processos para tornar mais efetivo o uso das pessoas que estão envolvidas no projeto.
- **Gerenciamento das comunicações do projeto** – Nessa área são executados os processos para a geração, coleção, disseminação, dissertação, armazenamento e disposição final de informação de projeto.

- **Gerenciamento de riscos do projeto** – Envolve processos relacionados à identificação e análise dos riscos, bem como o plano de resposta aos riscos, o qual visa contornar as ocorrências adversas.
- **Gerenciamento de aquisições do projeto** – Envolve processos necessários para adquirir bens e serviços externos à organização.

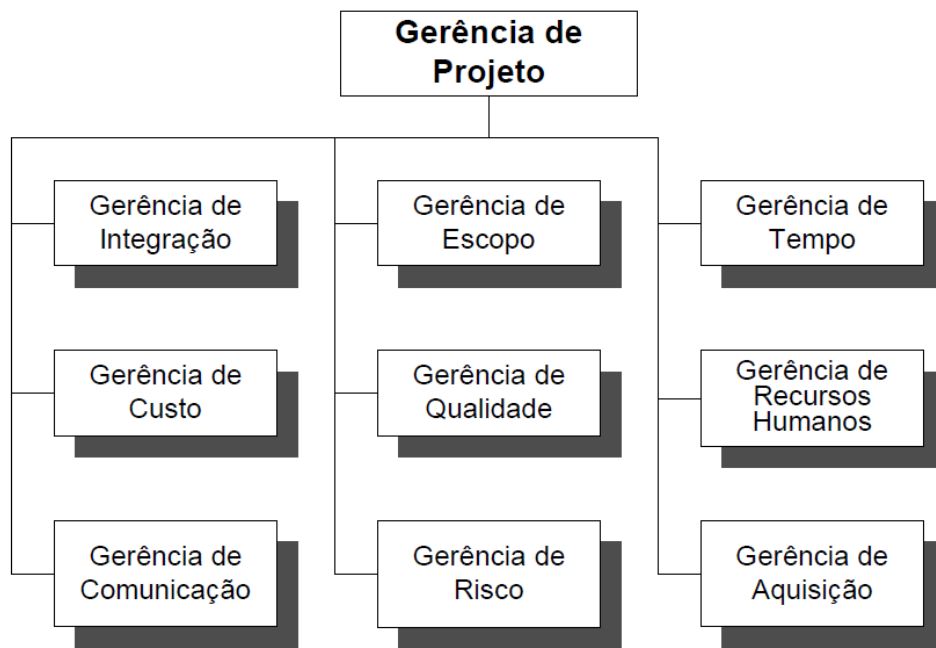


Figura 3 - Áreas de Conhecimento de Gerenciamento de Projetos

Fonte: Guia PMBOK. Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos. 4ª edição.

Nessas nove áreas de conhecimento, agrupam-se processos de gerenciamento de projetos, os quais são definidos por suas entradas, ferramentas, técnicas e saídas.

Os processos de gerenciamento de projetos são agrupados em cinco grupos: iniciação, planejamento, execução, monitoramento e controle e encerramento. Para

que se possa encerrar um projeto, primeiramente é necessário que este projeto tenha sido planejado, executado e monitorado.

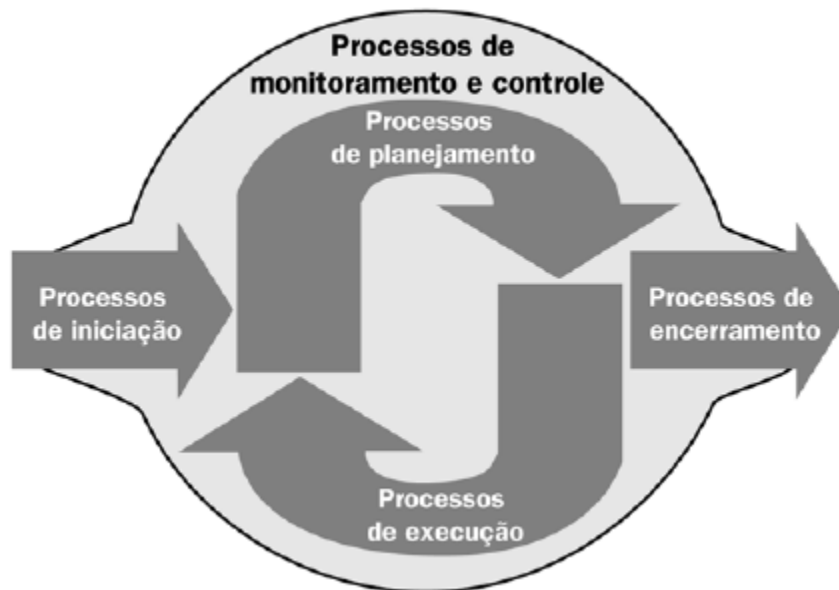


Figura 4 - Conexão entre os Grupos de Processos

Fonte: Guia PMBOK®. Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos. 4ª edição.

Processos de Iniciação – autorização do projeto ou de fase do projeto.

Processos de Planejamento – são processos de refinamento e definição de objetivos e escolha dos meios para atingir os objetivos.

Processos de Execução – execução do plano do projeto e gerenciamento dos recursos para desenvolvimento do plano.

Processos de Monitoramento e Controle – Têm como finalidade garantir que os objetivos do projeto são alcançados através do monitoramento e medição regular do progresso.

Processos de Encerramento – Processos executados para finalizar todas as atividades de todos os grupos de processos, visando encerrar formalmente o projeto ou a fase.

2.2 Encerramento de Projeto

A coleta de lições aprendidas pode ocorrer ao longo do projeto ou ao seu término, ou podem ser obtidas no processo de realização do projeto. As lições aprendidas podem ser identificadas a qualquer momento. Quando ocorre no encerramento de um projeto é mais conhecida como análise de Pós-Projeto ou *postmortem*.

O projeto atinge encerramento, quando todos os planos e processos incorporados foram finalizados ou o projeto foi cancelado. Nessa fase, os critérios para o sucesso do projeto são revisitados. Uma vez que o encerramento é estabelecido, arquivamentos, *postmortem* (retrospectiva), e atividades de melhoria de processos são realizadas (SWEBOK, 2004).

No PMBOK, os processos de encerramento pertencem às áreas de conhecimento integração e aquisições. Existem dois processos: Encerrar o projeto ou a fase e o encerramento de aquisições.

O processo de encerramento de projeto ou fase ocorre ao final do projeto, e inclui a finalização de todas as atividades pendentes em todos os processos de gerenciamento, de modo que o projeto seja formalmente encerramento ou cancelado. Nesse processo é elaborado um relatório de encerramento, que formaliza o fim do projeto para o *stakeholders* (interessados do projeto).

O diagrama apresentado na figura 5 apresenta as entradas, saídas e as ferramentas e técnicas utilizadas no encerramento.

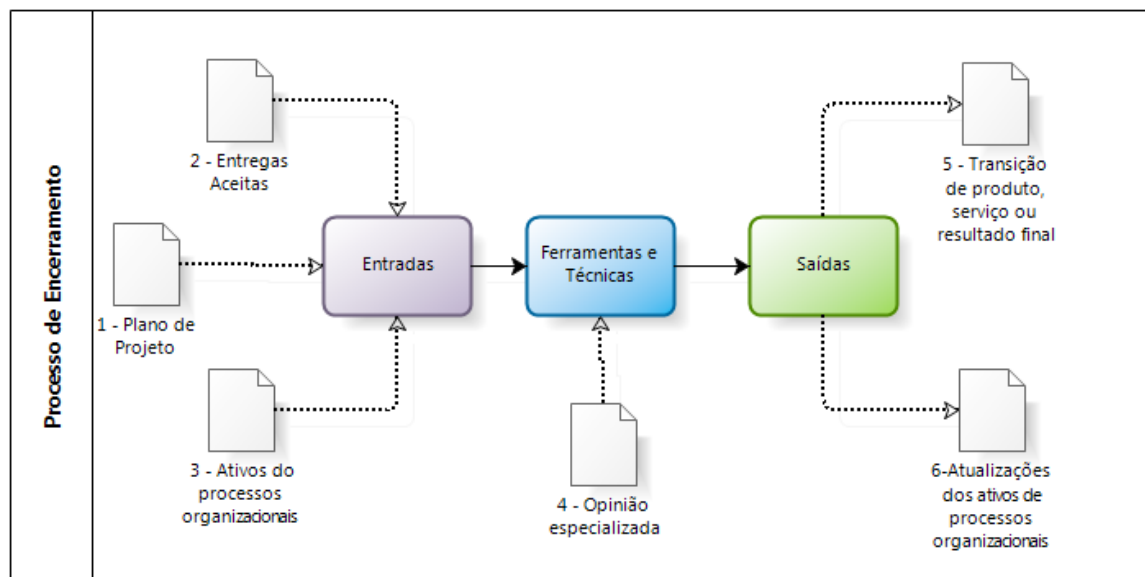


Figura 5 - Processo de Encerramento de Projeto

Fonte: Adaptação do Guia PMBOK. Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos. 4ª edição.

Quanto às técnicas para executar o processo de encerramento, possui como entrada a opinião especializada, a qual é fornecida baseada em especialização numa área de aplicação, área de conhecimento, disciplina, indústria, etc, adequada para a atividade que está sendo realizada (PMBOK, 2008). Essa especialização pode ser oferecida por qualquer grupo ou pessoa com formação, conhecimento, habilidade, experiência ou treinamento

No Encerramento, a equipe do projeto deve diagnosticar as lições aprendidas. As respostas vão permitir um melhor gerenciamento dos próximos projetos. Nesse processo, os documentos do projeto são armazenados para a aprendizagem contínua sobre o processo e o projeto de software.

2.3 Técnicas de Lições Aprendidas

Em publicações atuais, podem-se encontrar os mais variados modelos para a realização de uma análise mais detalhada no final de um projeto, documentar as

experiências positivas e negativas adquiridas no sentido de uma “análise de pós projeto“ (*postmortem*) ou uma "retrospectiva". No mesmo contexto, o termo "lições aprendidas" é frequentemente usado e é adotado neste trabalho como um termo superordenado.

Lições aprendidas são definidas como experiências chave de um projeto, as quais têm relevância para futuros projetos. São validadas pela equipe de projeto e representam um consenso sobre as percepções fundamentais que devem ser consideradas em projetos futuros (SCHINDLER, 2003).

No Encerramento de projeto, a equipe do projeto deve diagnosticar as lições aprendidas. As respostas vão permitir um melhor gerenciamento aos próximos projetos. Nesse processo, os documentos do projeto são armazenados para a aprendizagem contínua sobre o processo e o projeto de software.

O modelo de aprendizagem apresentado neste trabalho é o definido por Nonaka e Takeuchi em 2007, que classifica o conhecimento em dois tipos: o conhecimento tácito, e o conhecimento explícito. O conhecimento tácito é aquele que os seres humanos possuem, mas são incapazes de representar. Um exemplo é a ação de andar de bicicleta, a qual é difícil escrever. Já o conhecimento explícito pode ser representado como diretrizes de processo ou qualquer outra forma de documentação.

A documentação de lições aprendidas possui como objetivo: identificar os fracassos e sucessos de uma fase ou projeto e recomendações de melhoria dos processos.

2.3.1 Análise de Pós-Projeto

O objetivo da análise de pós-projeto ou de encerramento, é determinar o que deu certo, o que deu errado, o que funcionou, o que não funcionou, e como isso o que pode ser melhorado da próxima vez (DINGSØYR, 2001). Este tipo de aprendizagem pode ser apoiado através da análise de dados de projetos concluídos. A análise de *postmortem* é necessária para compreender o desempenho do processo sobre o projeto, e por conseguinte para determinar a capacidade do processo (JALOTE, 2002). A Figura 6 ilustra o papel da análise de pós-projeto .

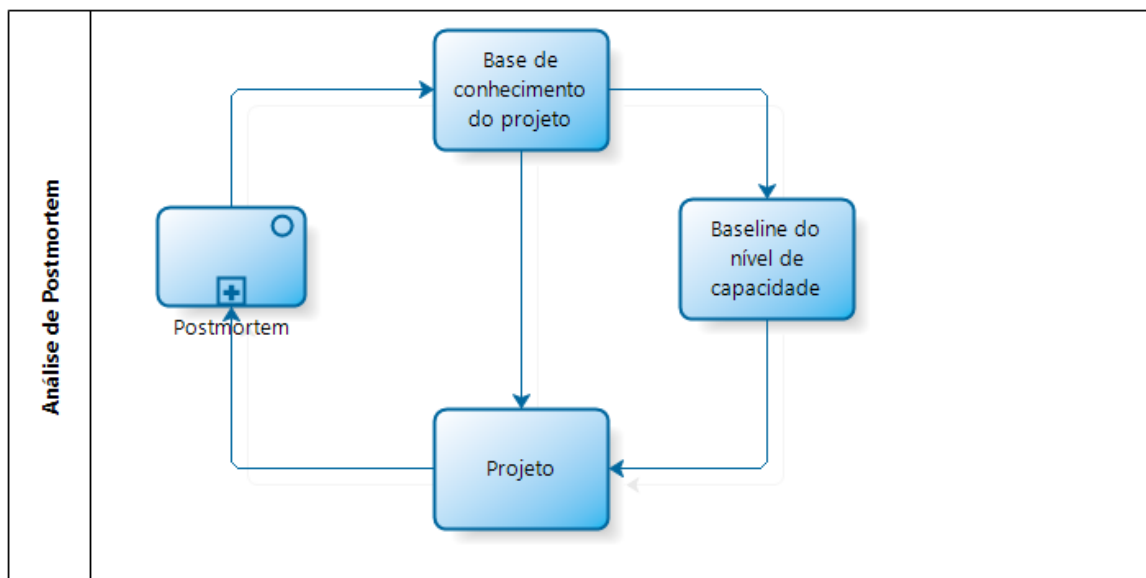


Figura 6 - Análise de *Pós-projeto*

Fonte : Adptação Software project management in practice / Pankaj Jalote p.234.

A análise de pós-projeto é composta por três fases: preparação, coleta de dados, e análise. Para cada fase, os membros da equipe podem aplicar uma série de métodos bastante simples, como o método KJ do etnólogo japonês Kawakita Jiro (DINGSØYR, 2001).

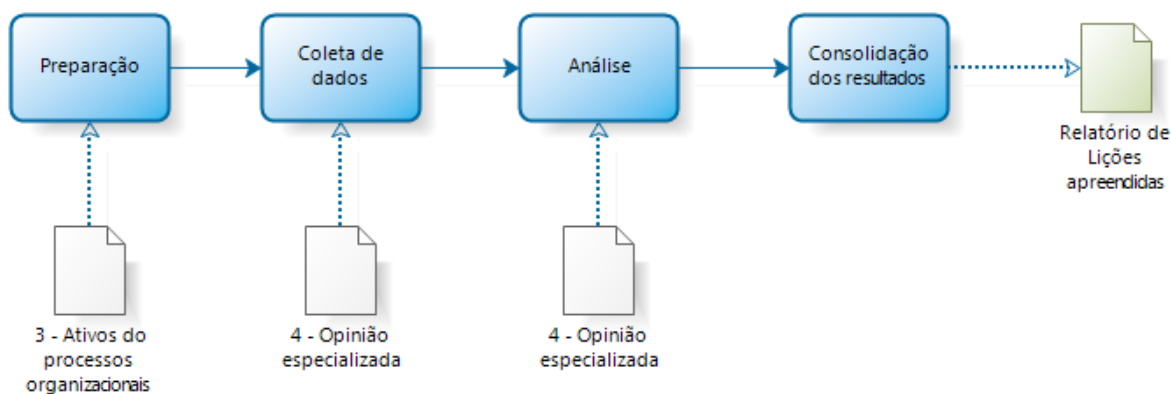


Figura 7 - Etapas da realização de PostmemJiro
 Fonte: produção da própria autora

Preparação - durante a fase de preparação, é feita uma revisão do histórico do projeto para entender melhor o que aconteceu. Todos os documentos disponíveis, tais como a estrutura de divisão de trabalho, planos de projeto, relatórios de revisão e relatórios do projeto são verificados.

Coleta de dados - são coletadas experiências através de uma sessão, derivada do método KJ ou diagrama de afinidades. O método KJ é utilizado para identificar problemas de uma maneira simples, que muitos conhecem por “Técnica do Papelzinho”. Os participantes recebem até cinco pequenos papéis lembrete (*post-its*), onde dão *feedback* sobre os eventos do projeto. Caso a reunião tenha sido formada por quatro pessoas, haverá vinte prováveis problemas para serem analisados. Em seguida, agrupam-se os papéis por afinidade do tema e monta-se uma seleção de assuntos.

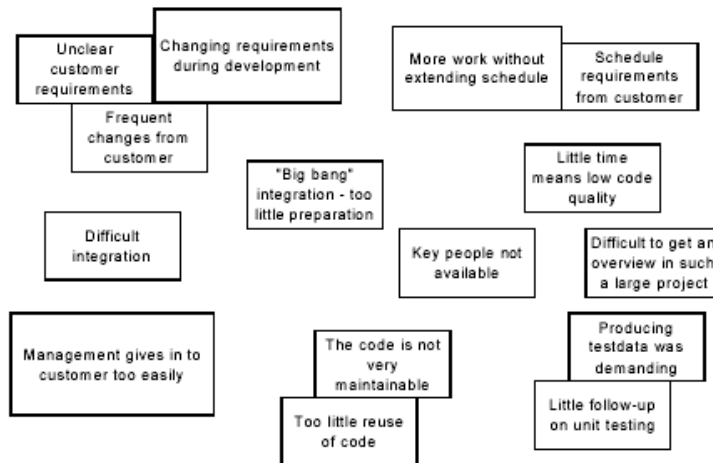


Figura 8 – Método KJ

Fonte: Dingsøy. Augmenting Experience Reports with Lightweight Postmortem Reviews. Conference on Product Focused Software Process Improvement, 2001.

Análise - são elaborados os diagramas Ishikawa, também conhecido como diagrama de causa e efeito, com o objetivo de levantar as causas que conduziram o projeto aos eventos identificados na sessão anterior. O diagrama Ishikawa permite estruturar hierarquicamente a causa raiz de determinado problema ou as oportunidades de melhoria, bem como os seus efeitos.

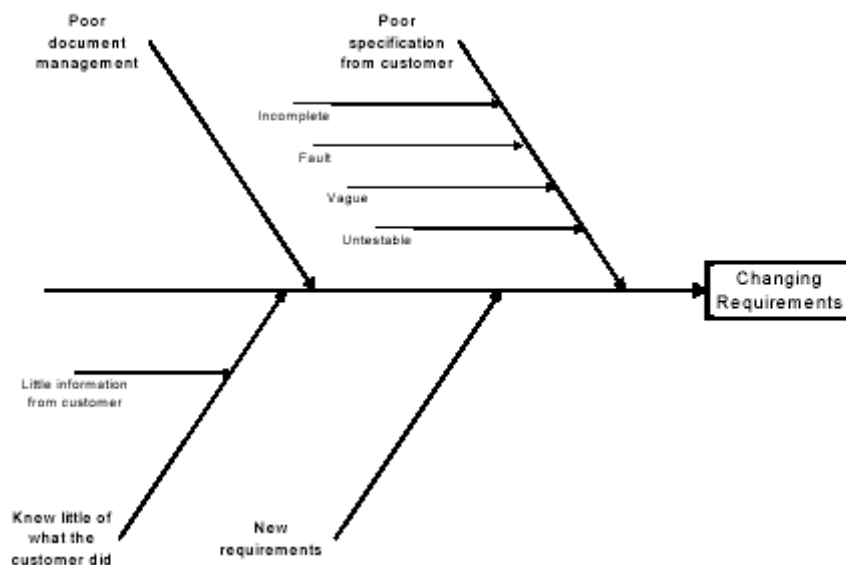


Figura 9 - Diagrama de Ishikawa

Fonte: Dingsøy. Augmenting Experience Reports with Lightweight Postmortem Reviews. Conference on Product Focused Software Process Improvement, 2001.

Consolidação dos resultados – é realizada documentação das lições aprendidas, o relatório dos resultados da análise de pós-projeto, o qual contém os seguintes tópicos:

- Introdução com a descrição do projeto
- Comparativo entre os resultados e o planejado;
 - Status
 - Equipe
 - Marcos
 - Custo
 - Qualidade
- Lições aprendidas
 - A documentação dos sucessos
 - O que não correu bem
 - Análise das técnicas utilizadas
- Recomendações

Os dados obtidos durante a análise de pós-projeto podem ser armazenados em uma base de conhecimento, através de pacotes de experiência. O objetivo de empacotar é documentar e partilhar as lições aprendidas, tornando a informação disponível para futuros projetos ou para o refinamento do próximo ciclo de melhoria. (JALOTE, 2002).

2.3.2 Oficina de Experiências

A oficina de experiências ou *Experince Factory* é um framework para melhoria de processos de engenharia de software, que visa evolução contínua de processos e artefatos de software (BASILI, 1994). A implementação deste framework consiste em

um repositório para disseminação de conhecimento de projetos de engenharia na intranet de uma empresa, visando a reutilização potencial por outros projetos.

O repositório é formado por pacotes de experiência, modelos e medidas de processos de software diferentes, produtos e outras formas de conhecimento através de pessoas, documentos e apoio automatizado. A eficácia do framework é revisado e medido contra com os objetivos do projeto e relatórios de encerramento do projeto e são publicadas no repositório.

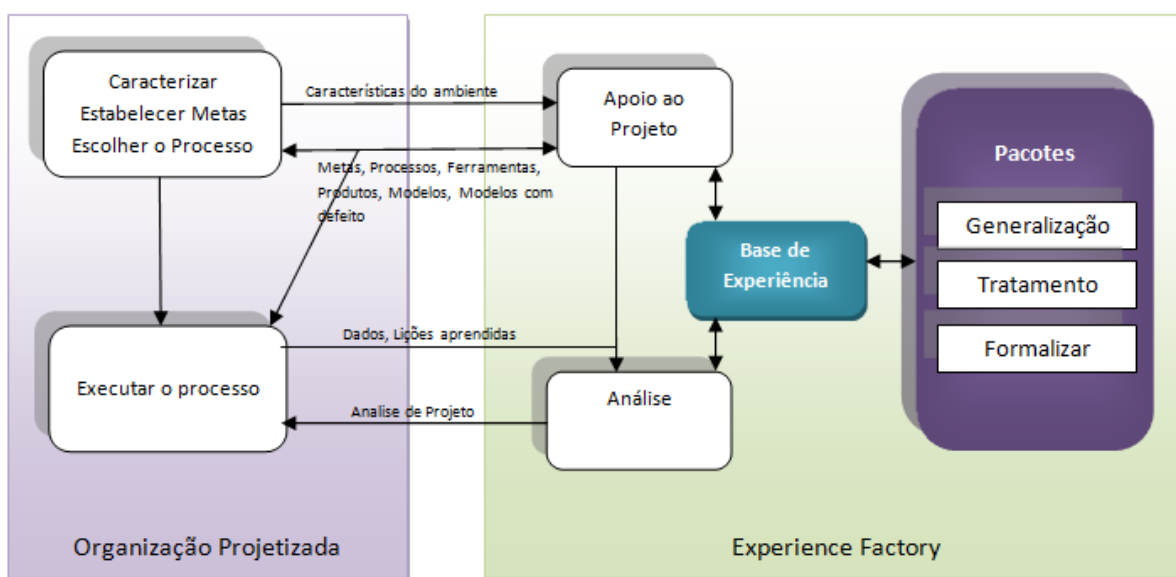


Figura 10 - Adaptação - Funcionamento de um Experience Factory.
 Fonte : Experience Factory. Encyclopedia of Software Engineering Volume
 1:469-476.

2.4 Modelos de Referência

A complexidade em arquiteturas de tecnologia da informação e infraestruturas é crescente. Por esse motivo, há muitos modelos de melhores práticas atualmente para a área da tecnologia da informação, os quais apoiam a implementação e melhoria dos processos. Desta maneira, foram eleitos para o desenvolvimento deste

trabalho um framework para gestão, qualidade, desenvolvimento e desenvolvimento ágil. A seguir, serão apresentados cada um destes frameworks.

	Tempo	1991	1994	1996	2001
Frameworks	Gestão			PMBOK PMI	
	Qualidade				CMMI SE
	Desenvolvimento		RUP Rational		
	Ágil	SCRUM Schwaber & Sutherland			

Figura 11 – Frameworks e sua data de lançamento
Fonte: produção da própria autora

2.4.1 PMI

Estabelecido em 1969 e sediado na Filadélfia, Pensilvânia EUA, o PMI - *Project Management Institute* é a principal associação mundial sem fins lucrativos em gerenciamento de projetos, atualmente com mais de 500 mil associados em 185 países. O PMI ocupa uma posição de destaque no desenvolvimento de padrões para a prática da profissão de gerenciamento de projetos no mundo. A publicação mais popular do PMI é o PMBOK - *A Guide to the Project Management Body of Knowledge*, aprovado como um ANS - Padrão Nacional Americano pelo ANSI - Instituto de Padrões Nacional Americano. É um padrão globalmente reconhecido para o gerenciamento de projetos.

O PMBOK Identifica na comunidade de gerenciamento de projetos um conjunto de conhecimentos amplamente reconhecidos como boas práticas, aplicáveis à maioria dos projetos na maior parte do tempo. Estes conhecimentos estão categorizados em nove áreas e os processos relacionados são organizados em cinco grupos de processos ao longo do ciclo de vida do projeto conforme apresentado na seção 1.1 deste trabalho.

Cada área de conhecimento abrange diversos processos no gerenciamento de projetos. Um processo é um conjunto de ações e atividades interrelacionadas que são executadas para alcançar um objetivo. Os processos caracterizam-se por suas entradas e saídas. Considerando-se estas ligações, podemos descrever cada processo em termos de:

- Entradas – documentos ou itens documentáveis que influenciarão o processo.
- Ferramentas e técnicas – mecanismos aplicados às entradas para criar as saídas.
- Saídas – documentos ou itens documentáveis resultantes do processo.

Os grupos de processos de gerenciamento de projetos estão ligados pelos objetivos que produzem. Em geral, as saídas de um processo tornam-se as entradas para outro processo ou são entregas do projeto.

2.4.2 CMMI

O CMMI - *Capability Maturity Model Integration* é um modelo de referência para a Engenharia de Software, formulado pelo *Software Engineering Institute da Carnegie-Mellon University*. O CMMI foi encomendado e patrocinado pelo Departamento de Defesa dos Estados Unidos, com o propósito de avaliar a capacidade de seus

fornecedores de *software*. O CMMI tem grande aceitação da indústria mundial e suas práticas podem ser aplicadas ao desenvolvimento de produtos e serviços e não somente a softwares. No modelo CMMI, utilizam-se níveis para descrever um caminho evolutivo recomendado para uma organização que deseja melhorar os processos utilizados para desenvolver e manter seus produtos e serviços (CMMI-DEV 1.2 p.47). A versão atual do CMMI 1.3 foi publicada em 2010 apresenta três constelações:

- *CMMI for Development* (CMMI-DEV), voltado ao processo de desenvolvimento de produtos e serviços.
- *CMMI for Acquisition* (CMMI-ACQ), voltado aos processos de aquisição e terceirização de bens e serviços.
- *CMMI for Services* (CMMI-SVC), voltado aos processos de empresas prestadoras de serviços.

Esse trabalho tem o foco em desenvolvimento de software. Assim sendo, o CMMI-DEV é explorado. O CMMI-DEV possui duas formas de representação:

Por Estágios: a empresa busca avaliar a sua maturidade. Em cada nível de maturidade a organização precisa implementar determinadas áreas de processos. Busca aumentar a maturidade da organização. Aumenta o nível de capacidade. Os Níveis de Capacidade vão de 0 a 5;

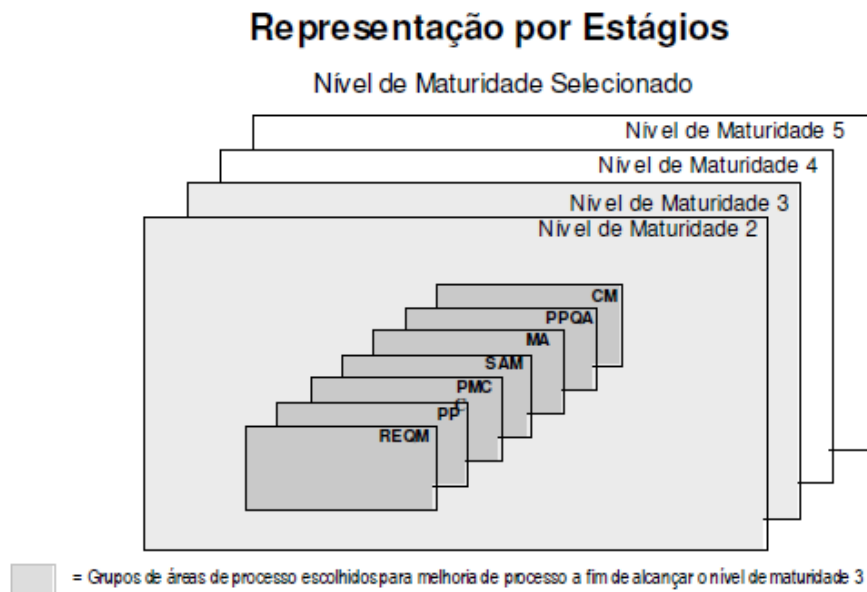


Figura 12 - Representação por estágios
Fonte: CMMI-DEV 1.2

Contínuo: ao invés de implementar um conjunto de áreas de processos, a organização escolhe quais áreas são estratégicas para ela. Busca aumentar a capacidade em algumas áreas. Os níveis de Maturidade vão de 1 a 5; Essa representação é mais utilizada.

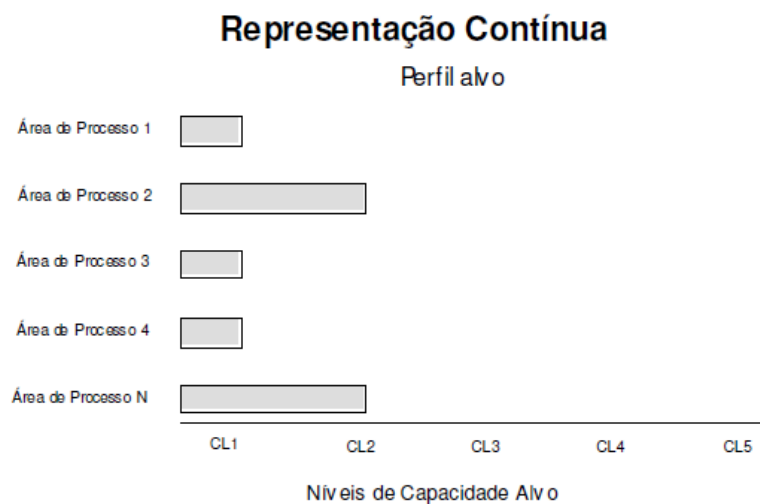


Figura 13 - Representação contínua
Fonte: CMMI-DEV 1.2

Com a premissa: "A qualidade é influenciada pelo processo", o objetivo do CMMI é fornecer orientações para melhorar os processos de uma organização e sua capacidade para gerir desenvolvimento, aquisição e manutenção de produtos ou serviços. A versão atual do modelo - CMMI-DEV 1.3 - possui 22 áreas de processo, abaixo algumas dessas áreas de processo:

Planejamento de Projeto – Visa a construção do plano de projeto, o qual engloba as definições das atividades a serem executadas.

Monitoramento e Controle de Projeto – Possui como objetivo fornecer uma compreensão do andamento do projeto para que ações corretivas apropriadas possam ser tomadas quando o andamento do projeto se desvia do plano.

Gestão Integrada de Projeto - Visa o estabelecimento e gerenciamento do envolvimento das partes interessadas no projeto, de acordo com um processo integrado e definido que é adaptado da organização do conjunto de processos padrão.

Gestão de Riscos - Identifica potenciais problemas antes que eles ocorram, de modo a planejar atividades que possam ser necessárias durante o projeto para mitigar os impactos negativos sobre a especificação dos objetivos.

Gestão Quantitativa de Projeto – Possui como objetivo gerir quantitativamente o processo de definição do projeto.

Gestão de Contrato com Fornecedores (SAM) - Gestão de Contrato com Fornecedores é uma área de processo de gerenciamento de projeto do nível de maturidade 2. O objetivo da Gestão de Contrato com Fornecedores (SAM) é gerenciar a aquisição de produtos e serviços de fornecedores. Tais aquisições podem ser utilizadas para desenvolver e manter o produto ou o serviço, e

principalmente, tratam dos entregáveis que serão repassados no encerramento ao cliente do projeto.

Geralmente, os produtos a serem adquiridos pelo projeto são determinados durante as fases iniciais de planejamento e desenvolvimento do produto (CMMI) e a área Gestão de Contrato com Fornecedores (SAM) contempla as seguintes práticas:

- SG 1 Estabelecer Acordos com Fornecedores
 - SP 1.1 Determinar Tipo de Aquisição
 - SP 1.2 Selecionar Fornecedores
 - SP 1.3 Estabelecer Acordos com Fornecedores
- SG 2 Satisfazer o Acordo com Fornecedor
 - SP 2.1 Executar Acordo com Fornecedor
 - SP 2.2 Aceitar Aquisições
 - SP 2.3 Garantir a Transferência de Produtos

O CMMI não descreve como as organizações devem realizar o SAM, ele apenas aborda as melhores práticas para o gerenciamento de projetos de software sem um método, técnica ou ferramenta específica.

2.4.3 Projetos de Software Agile

Devido a complexidade, tamanho, mudanças de requisitos, urgência e necessidade de demonstrar valor mais rápido, os modelos de ciclo de vida de desenvolvimento de software de maneira interativa e incremental ganham cada vez mais espaço. Com o lema: dividir para conquistar, onde o software é construído em partes, ou seja, em ciclos também chamado de iterações, a cada iteração é feito um novo incremento, parte do software funcional, até completar o software.

“Métodos ágeis” é o nome dado a um conjunto de práticas que pretendem conduzir o processo de desenvolvimento de software com agilidade. Em 2001, um grupo de desenvolvedores e engenheiros de software considerava o modelo de desenvolvimento em cascata muito burocrático, pesado e inconsistente, esse grupo desenvolveu e publicou o Manifesto Ágil. (MANIFESTO, 2011). Em sua publicação, o manifesto considera a valorização de:

- Indivíduos e interações mais que processos e ferramentas;
- Software funcionando mais que documentação abrangente;
- Colaboração do cliente mais que longa negociação do contrato;
- Responder a mudanças mais que seguir um plano;

No foco dos princípios a serem seguidos no manifesto os aspectos humanos são mais importantes: a motivação da equipe e a frequente interação de patrocinadores e cliente, a entrega de um produto que atenda à necessidade do cliente, planejamento e documentação devem evitar os excessos e as redundâncias.

A seguir, será apresentado o framework de desenvolvimento ágil SCRUM desenvolvido por Ken Schwaber e Jeff Sutherland.

2.4.4 SCRUM

Scrum é um framework para desenvolver e manter produtos complexos (SCHWABER & SUTHERLAND, 2011), focado no gerenciamento de projetos para pequenas equipes. SCRUM é definido por um processo de desenvolvimento de sistemas como um conjunto de atividades dispersas que combinam: ferramentas e técnicas viáveis com a melhor equipe de desenvolvimento possível. Uma vez que estas atividades estão dispersas, os controles para gerenciar o processo e os riscos

inerentes são usados. SCRUM é uma valorização do uso do ciclo de desenvolvimento iterativo / incremental orientada a objetos (SCHWABER, 1995).

Conforme a Scrum Alliance, o framework da metodologia Scrum pode ser definido com os seguintes componentes:

Alguns conceitos:

- *Sprint*: é a base do Scrum, possui uma interação de 2 a 4 semanas.
- *Release*: são os itens do backlog que constituem uma entrega viável, considerando as variáveis de tempo, qualidade e competitividade.
- *Pacotes*: partes de produtos que devem ser trabalhados para implementar um item de *backlog*. Este pacote pode ser constituído de documentação, diagramas e códigos fonte.
- *Equipes Scrum*: são os indivíduos responsáveis por implementar os itens de *backlog* na nova release.

Três Papéis:

- *Scrum Master* : O facilitador da equipe, ele mantém a equipe focada e é o mantedor das praticas do Scrum;
- *Scrum Time*: é uma equipe auto gerenciável, multifuncional, multidisciplinar, formada por menos de 10 pessoas.
- *Dono do Produto*: representante do cliente, mantém a lista de prioridades, ajuda no entendimento do que deve ser feito, cria metas e objetivos.

Artefatos:

- *Product Backlog*: é lista de requisitos funcionais do produto. Os bugs, defeitos e as solicitações de melhoria.

- *Sprint Backlog*: são os requisitos divididos em tarefas.
- *Burndown chart*: gráfico de Horas estimadas em razão do tempo planejado.

Cerimônias:

- *Planejamento do Sprint*: equipe seleciona itens do Product Backlog com os quais se compromete em concluir. Apresenta-se as histórias, quebra-se as histórias em tarefas. Define-se o *Sprint Goal* e o *Sprint Backlog*.
- *Stand-up Meeting*: reunião diária com duração de no máximo 15 minutos, o Scrum Team segue o seguinte script: –O que fiz ontem? –O que farei hoje? –Tem algum obstáculo?
- *Sprint Review*: Nesta cerimônia são apresentação das funcionalidades do sistema para o dono do produto.
- *Sprint Retrospective*: Reunião de lições aprendidas após a entrega das funcionalidades ao dono do produto.

O *Scrum* pode ser dividido em três fases distintas. Esse conjunto de três fases que constituem um ciclo de desenvolvimento é chamado de *sprint*. Esses *sprints* devem ter duração sugerida de até trinta dias e deverão gerar uma release. As fases são (ALLIANCE, 2010).

- *Planejamento*: definição da nova release baseada no *backlog* atualmente conhecido.
- *Desenvolvimento*: implementação das funcionalidades da nova release, respeitando-se as variáveis de tempo, requisitos qualidade e concorrência. A interação entre essas variáveis define o final do ciclo.

- Encerramento: preparação para o final da release, incluindo documentação e produto testado.

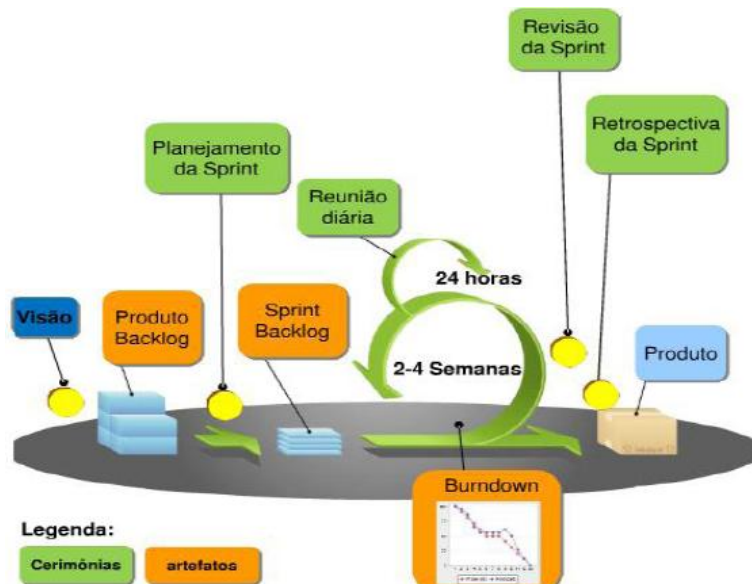


Figura 14 - Ciclo da metodologia scrum – sprint
Fonte: Scrum Alliance (adaptado)

Sprint Retrospective

Ao final de cada *sprint* e entrega do produto ao “dono do produto” é realizada a reunião de *sprint retrospective*. Toda a equipe incluído o *scrum master*. A equipe se reúne para discutir o que correu bem e avaliar o que deve melhorar no próximo *sprint*.

O objetivo da retrospectiva é verificar a forma como o *sprint* foi realizado no que diz respeito às pessoas, relações, processos e ferramentas (SCHWABER & SUTHERLAND, 2011). A inspeção deve identificar e priorizar os principais itens que correrão bem e os itens que, se feito de forma diferente, poderiam tornar as coisas ainda melhor. Estes incluem *scrum team* composição, arranjos de reuniões, ferramentas, definição de "pronto", métodos de comunicação e processos para transformar itens *product backlog* em algo "feito". Até o final da Retrospectiva da

sprint, o scrum team deve ter identificado melhorias acionáveis e medidas que implementa no próximo. Essas mudanças se tornam a adaptação para a inspeção empírica.

"A reunião de Sprint Retrospective é uma parte integrante do processo de inspeção e adaptação, caso contrário, a equipe nunca será capaz de melhorar a sua produção global e não o foco no desempenho geral da equipe. O ScrumMaster deve prestar atenção a esta reunião e trabalhar para resolver os impedimentos que podem estar desmotivando a equipe." Kelley Louie (Certified Scrum Practitioner).

Uma abordagem adotada é que todo o time se reúna seguindo o seguinte script: Começar a fazer; Parar de fazer; Continuar fazendo.

2.4.5 RUP

RUP (Rational Unified Process) é um processo definido pela Rational para a gestão de projetos de desenvolvimento de software. É construído sobre as boas práticas encontradas no mercado. O RUP utiliza a UML (Unified Modeling Language) como ferramenta para a especificação do sistema e dos testes (MARTINS, 2006). É um processo forte, isso é, contém fases, disciplinas, papéis e artefatos bem definidos e requer uma efetiva participação de um gerente. Entre as características fundamentais do RUP: Dirigido por casos de uso, centrado na arquitetura, e iterativo e incremental.

A divisão de um projeto em partes entregáveis é uma prática comum no RUP, onde cada miniprojeto desses equivale a uma iteração que resulta em um incremento. As iterações são fases do projeto, e os incrementos são as evoluções do produto, onde cada iteração dá continuidade ao trabalho desenvolvido na iteração anterior. O resultado de cada miniprojeto é um executável contendo os

requisitos que foram selecionados e planejados. As fases da iteração podem ser visualizadas no famoso “gráfico de baleias” como é apresentado na figura abaixo.

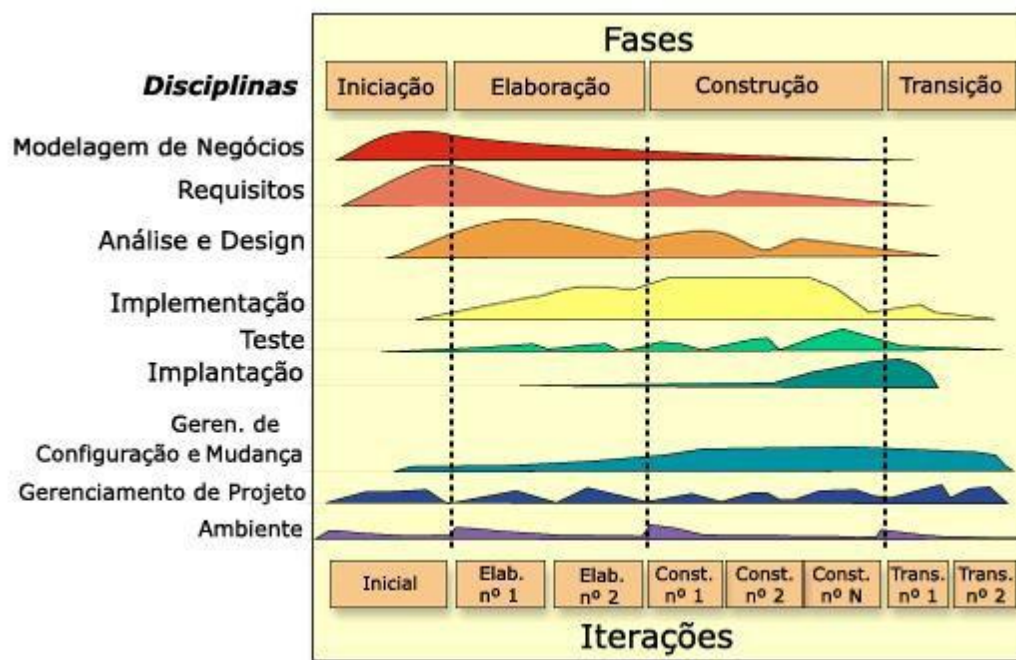


Figura 15 - Gráfico das iterações do RUP

Um dos pilares deste framework é o conceito de *best practices* (melhores práticas), para tornar o desenvolvimento mais eficiente. O RUP define seis *best practices*, sendo elas:

- **Desenvolvimento iterativo:** trata-se do desenvolvimento em ciclos, onde cada ciclo possui um objetivo a ser alcançado. Esta prática possibilita a identificação e modificação de requisitos com maior flexibilidade; integração progressiva de elementos ao software, ocasionando uma melhora no gerenciamento de riscos.
- **Gerenciamento de requisitos:** uma maneira prática de criar, organizar, comunicar e organizar os requisitos do projeto. O RUP é uma metodologia dirigida-a-casos-de-uso (use-driven-case), de modo que é possível utilizar os

mesmos casos de uso definidos no sistema como base para o resto do processo.

- **Arquitetura baseadas em componentes:** desenvolvimento na modularizado, através do uso de componentes, de modo a criar um sistema flexível, adaptável, intuitivamente entendível e reutilizável. Entre os benefícios há a facilidade para identificar, isolar, manipular e desenvolver componentes é maior do que para um sistema inteiro.
- **Modelagem visual:** permite melhor entendimento não somente da concepção e a complexidade do sistema, mas também mensurar a dimensão do problema, além de facilitar a identificação da solução.
- **Verificação continua da qualidade:** no RUP a qualidade é responsabilidade de todos os membros da equipe. A qualidade é focada especialmente em duas áreas:
 - Qualidade de produto: a qualidade do produto sendo desenvolvido (software ou sistema) e todas as partes envolvidas (componentes, subsistemas, arquitetura, etc).
 - Qualidade de processo: a qualidade dos processos dentro do projeto de desenvolvimento.
- **Controle de mudanças:** muitas mudanças ocorrem no decorrer de um projeto como resultado de um processo de desenvolvimento iterativo. Controlar as mudanças durante todo o projeto é fundamental para manter a qualidade do projeto.

No RUP os processos de desenvolvimento são dividido em ciclos, sendo que o ciclo de desenvolvimento é subdividido em 4 fases consecutivas. Estas fases são

concluídas tão logo uma *milestone* é alcançada. Uma *milestone* define uma etapa, na fase, na qual decisões críticas são feitas ou objetivos são alcançados.

- **Concepção:** é a fase aonde é feita uma discussão sobre o problema, definição do escopo do projeto, estimativa de recursos necessários para a execução do projeto, etc. É nesta fase que é apresentado o plano de projeto, caso de uso inicial e o glossário do projeto, entre outros.
- **Elaboração:** o propósito desta fase é analisar o domínio do problema, desenvolver o plano de projeto, estabelecer a arquitetura e eliminar os elementos de alto risco. Esta é a fase mais crítica, pois ao final desta a engenharia é considerada completa e os custos para modificação do sistema aumentam a medida que o projeto avança.
- **Construção:** Esta fase compreende a fase de modelagem e a fase de desenvolvimento. A fase de modelagem deve utilizar alguma notação definida pela UML.
- **Transição:** A partir desta fase, o sistema já está pronto, começa a implantação do sistema. Nesta fase é deve ser utilizado o lançamento de versões beta, operação paralela com o sistema legado, treinamento dos usuários e mantenedores do sistema, etc.

Objetivo do RUP é assegurar uma produção de alta qualidade de software, que realiza a necessidade do usuário seguindo prazos e o orçamento. O RUP, como processo de desenvolvimento de software, tem 4 regras: servir de guia; especificar quais artefatos devem ser desenvolvidos e quando devem ser desenvolvidos; dirigir as tarefas individuais e do time como um todo; oferecer critérios para monitorar e medir os produtos e atividades do projeto.

2.4.6 Papeis e Atividade de uma Organização de Software

O processo a ser definido deverá ser aplicável em uma entidade de software. Como base da descrição dos papéis e responsabilidades será seguido as orientações do modelo RUP.

Um papel pode ser desempenhado por uma pessoa ou um grupo de pessoa. Um membro da equipe do projeto geralmente desempenha muitos papéis distintos. Os papéis não são pessoas; pelo contrário, eles descrevem como as pessoas se comportam no negócio e quais são as responsabilidades que elas têm (RUP, 2001).

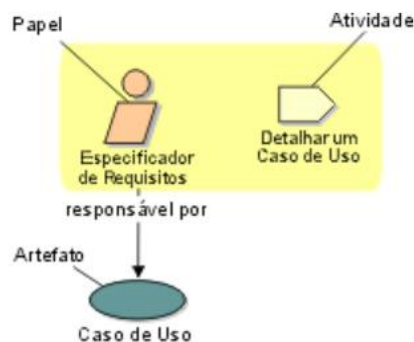


Figura 16 - Um papel, suas atividades e artefatos

Os papéis têm um conjunto de atividades coerentes por eles executadas. As atividades estão fortemente relacionadas aos artefatos. Os artefatos fornecem a entrada e a saída para as atividades e o mecanismo pelo qual as informações são transmitidas entre as atividades (RUP, 2001).

CAPÍTULO 3

ESTADO DA ARTE

Este capítulo apresenta a análise do estado da arte, a definição de um processo genérico para o gerenciamento de lições aprendidas. Possui como objetivo uma modelagem genérica, alinhada com os principais frameworks de melhores práticas de controle e gerência de TI, com base na revisão bibliográfica realizada na etapa anterior deste trabalho.

3.1 Análise do Estado da Arte

Este tópico apresenta um mapeamento do gerenciamento de lições aprendidas orientadas pelos modelos: PMBOK, CMMI, RUP e SCRUM. O mapeamento objetiva investigar as similaridades e as divergências entre as abordagens desses modelos.

A análise de gestão de lições aprendidas tomou como base neste trabalho o conceito de Angus e Patel (1998), que define a gestão de conhecimento em quatro processos:

- **Coleta** : Consiste na introdução de informações e dados em um sistema.
- **Processamento**: Consiste em adicionar valor às informações coletadas, tornando-as conhecimento.
- **Armazenagem**: Consiste em facilitar a localização, disponibilidade e utilização do conhecimento.
- **Distribuição**: que consiste em levar conhecimento às pessoas que precisam usá-lo, pois seu valor decorre do uso.

Com base nesse modelo, a seguir, serão analisados os frameworks escopo deste trabalho. Analogamente os processos para gestão do conhecimento passaram a ser as fases para gestão de lições aprendidas neste trabalho.

3.1.1 Lições Aprendidas no PMI

O Guia PMBOK considera as lições aprendidas como: a aprendizagem obtida durante todo o processo de realização do projeto e/ou prestação de serviços. As lições aprendidas documentadas fazem parte dos “ativos de processos organizacionais” da empresa.

Para esse modelo, no mínimo, as lições aprendidas devem ser coletadas durante o encerramento do projeto, o guia orienta a realização de uma revisão de pós-projeto e a documentação das lições aprendidas, porém não é explícito como devem ser realizadas essas atividades. As descrições destas atividades fazem parte da fase de encerramento de projeto.

Tabela 1 - Análise de Lições aprendidas no PMI

1 Coleta
<ul style="list-style-type: none"> Fazer uma revisão pós-projeto ou de final de fase.
2 Processamento
<ul style="list-style-type: none"> Documentar as lições aprendidas.
3 Armazenagem
<ul style="list-style-type: none"> Armazenamento no Ativos de Processos Organizacionais.
4 Distribuição
-

3.1.2 Lições Aprendidas no CMMI-DEV

Para o CMMI-DEV as lições aprendidas fazem parte dos ativos de processo da organização. Através da área de processo de *Análise e Resolução de Causas (CAR)*, as lições aprendidas são abordadas como o resultado de uma análise causa e efeito. Os defeitos e problemas podem ter sido previamente encontrados em outros projetos ou em fases ou tarefas iniciais de um projeto, as atividades de análise e resolução de causas podem ser vistas como um mecanismo de comunicação de lições aprendidas entre projetos.

A maneira de realizar a coleta de lições aprendidas é vista na área de processo *Foco nos Processos da Organização (OPF)* que expõe como devem ser realizados o planejamento, implementação e melhorias dos processos organizacionais. Isto é realizado com base na compreensão dos pontos fortes e pontos fracos dos processos e na análise dos ativos de processo da organização. Na subprática *SG 3 - Implantar os Ativos de Processo da Organização e Incorporar Lições Aprendidas*, os ativos de processo da organização são implantados na organização e as experiências relacionadas a processo são incorporados a base de conhecimento. E mais precisamente na prática específica *SP 3.4 - Incorporar Experiências Relacionadas a Processo nos Ativos de Processo da Organização*, é detalhando como tratar a base de conhecimento de projeto: deve-se incorporar aos ativos de processo da organização, os produtos de trabalho, as medidas e as informações para melhoria relacionada a processo que foram derivados do planejamento e da execução dos processos.

Por fim, as lições aprendidas são citadas como um dos métodos para identificação de riscos, durante o processo *Gestão de Riscos (RSKM)*. A *Gestão de Riscos* possui como objetivo identificar potenciais problemas antes que eles

ocorram, de modo que as atividades de contorno de riscos possam ser planejadas, postas em prática quando necessário, mitigando deste modo os impactos indesejáveis que afetam o andamento do projeto.

Tabela 2 - Análise de Lições aprendidas no CMMI

1 Coleta
<ul style="list-style-type: none">• Conduzir revisões periódicas da efetividade e da adequação do conjunto de processos-padrão da organização e de seus ativos de processo em relação aos objetivos estratégicos da organização.
2 Processamento
<ul style="list-style-type: none">• Identificar lições aprendidas a partir da definição, realização de pilotos, implementação.• Resultado de uma análise causa e efeito.
3 Armazenagem
<ul style="list-style-type: none">• Atualização dos ativos de processo da organização.
4 Distribuição
<ul style="list-style-type: none">• Tornar disponíveis as lições aprendidas às pessoas da organização conforme apropriado.

3.1.3 Lições Aprendidas no RUP

O RUP é um modelo de processo iterativo e incremental. Apesar de sua versão padrão não tratar especificamente de oportunidades de aprendizado, no final de cada interação há um o momento para a realização formal das atividades de aprendizado. O RUP prevê o debate sobre lições aprendidas através da revisão de

pós-projeto e ações corretivas no fechamento de cada iteração, ou seja, ocorre diversas vezes em um mesmo projeto.

O modelo considera a revisão pós-projeto para determinar se o projeto está pronto para a aceitação formal e definitiva do cliente e para a finalização subsequente. Essa revisão é realizada durante a *Avaliação de Iteração*, onde é levantada a lista de problemas, as quais são revisadas para assegurar que quaisquer questões remanescentes serão compreendidas e resolvidas, é gerado uma avaliação de status que captura os resultados da revisão pós-projeto a qual faz parte da aceitação final do projeto.

Tabela 3 - Análise de Lições aprendidas no RUP

1 Coleta
<ul style="list-style-type: none"> • Uma reunião post-mortem do projeto.
2 Processamento
<ul style="list-style-type: none"> • Desenvolvimento de um relatório post-mortem do projeto.
3 Armazenagem
-
4 Distribuição
-

3.1.4 Lições Aprendidas no SCRUM

No modelo ágil SCRUM há o controle do andamento do projeto, o encerramento de cada fase e as lições aprendidas. Na reunião de revisão do sprint (*Sprint Review*), a qual ocorre no final de cada *sprint*, é realizada uma revisão do trabalho que foi desenvolvido durante o projeto. O *feedback* é obtido de forma

colaborativa e informal, foca nos objetivos no trabalho e os passos da próxima interação. Nessa reunião é também realizada uma apresentação das funcionalidades do sistema para o dono do produto.

Na reunião de retrospectiva (*retrospective*) é realizada a revisão do processo de trabalho. É realizada o levantamento das lições aprendidas, essa reunião foca o framework e as práticas do Scrum, para se obter um melhor desempenho na próxima iteração. De acordo com Ken e Colin (2009), essa é uma reunião que permite a equipe do projeto discutir o sprint concluído de maneira a expressar o que poderia ser mudado, em relação a comunicação, o ambiente de trabalho, os artefatos e as ferramentas que são utilizadas durante o projeto. A reunião de retrospectiva objetiva a busca de uma melhor alternativa para tornar o próximo sprint mais agradável e produtivo.

Tabela 4 - Análise de Lições aprendidas no SCRUM

1 Coleta
<ul style="list-style-type: none"> Na Reunião de Retrospectiva (Retrospective) é realizada a revisão do processo de trabalho. Realizado o levantamento das lições aprendidas, essa reunião foca o framework e as práticas do Scrum.
2 Processamento
-
3 Armazenagem
-
4 Distribuição
-

Com base na análise realizada segue o comparativo dos modelos.

Tabela 5 – Comparativo dos modelos para Gerenciamento de Lições Aprendidas

PMI	CMMI-DEV	RUP	SCRUM
1. Coleta			
Realizar uma revisão pós-projeto ou de final de fase.	Conduzir revisões periódicas da efetividade e da adequação do conjunto de processos-padrão da organização e de seus ativos de processo em relação aos objetivos estratégicos da organização.	Uma reunião post-mortem do projeto	Na Reunião de Retrospectiva (Retrospective) é realizada a revisão do processo de trabalho. Realizado o levantamento das lições aprendidas.
2. Processamento			
Documentar as lições aprendidas.	Identificar lições aprendidas a partir da definição. Resultado de uma análise causa e efeito.	Desenvolvimento de um relatório post-mortem do projeto	-
3. Armazenagem			
Armazenamento no Ativos de Processos Organizacionais.	Atualização dos ativos de processo da organização.	-	-
4. Distribuição			
-	Tornar disponíveis as lições aprendidas às pessoas da organização conforme apropriado.	-	-

3.2 Definição do Processo Genérico

Nesta seção, é explicitado como foi realizada a elaboração do modelo de processo genérico de lições aprendidas. O processo foi concebido de acordo com os seguintes passos:

1. **Análise dos modelos e processos existentes:** na seção anterior foi realizada uma análise dos processos existentes para o gerenciamento de lições aprendidas, conforme os modelos disponíveis e escopo deste trabalho. Foi realizada a comparação dos termos de referencia quanto ao tema lições aprendida.

2. **Concepção de um de ciclo de vida a partir de um modelo previamente existente:** o modelo eleito para ser utilizado como base do processo de gerenciamento de lições aprendidas é o de Angus e Patel (1998) que divide analogamente a gestão do conhecimento nos processos de: coleta, processamento, armazenagem e distribuição.

3. **Para cada fase será dividida em atividades:** serão definidas as características como entradas, saídas, responsabilidades e objetivos de cada fase. Cada fase apresentará a tabela 6, a qual apresenta a estrutura a ser utilizada para apresentar as características de cada fase.

Tabela 6 – Modelo de Descrição das Características

Nome	Nome da Fase
Objetivos	Objetivos a serem atingidos na fase.
Papéis e Responsabilidades	Listagem dos papéis que participam da fase e suas contribuições.

Atividades	Atividades a serem realizadas para atingir o objetivo proposto.
Entradas	Informações e ações necessárias para que se possam iniciar as atividades da fase.
Saídas	Entregáveis (documentos) produzidos na fase.

A primeira etapa da concepção do processo genérico foi realizada no decorrer dos tópicos anteriores deste trabalho. A seguir, serão definidas as fases do processo com suas respectivas orientações.

3.3 Fases do Processo

O ciclo de vida do processo foi baseado na proposta de Angus e Patel para gestão do conhecimento, tal proposta compõem-se de quatro fases:

- **Coleta** – que consiste na introdução de informações e dados em um sistema, de forma qualitativa, organizada, sistemática, definitiva e dirigida a um objetivo.
- **Processamento** – trata-se de adicionar valor às informações coletadas, tornando-as conhecimento. Para isso, deve ser feita uma avaliação da adequação, da qualidade e do uso potencial das informações; além de interpretá-las, dividi-las, associa-las, resumi-las e amplia-las.
- **Armazenagem** – proporcionar a facilidade de localização, disponibilidade e utilização do conhecimento.
- **Distribuição** – que consiste em levar conhecimento às pessoas que precisam usá-lo, pois seu valor decorre do uso.

Esse conceito de composição da Gestão do Conhecimento será utilizado como as fases do processo de coleta de lições aprendidas nesse trabalho, abaixo a figura apresenta as fases do processo a ser concebido.

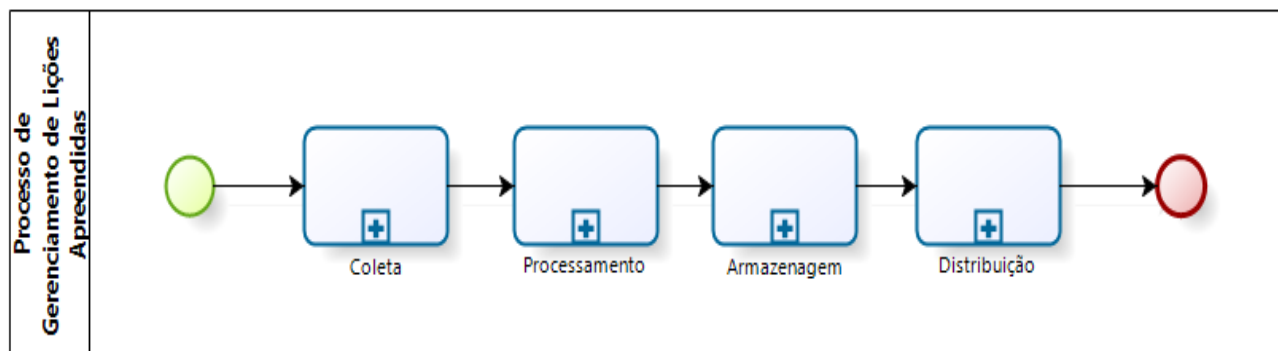


Figura 17 - Fases do Processo Proposto
Fonte: Produção da autora

A seguir, serão descritas as fases do processo genérico com base na referência bibliográfica e análise do estado da arte, utilizando a tabela de características proposta anteriormente.

3.3.1 Coleta

Coletar lições aprendidas é uma parte integrante de todo projeto e serve a vários propósitos. A formalização das lições aprendidas pode ser realizada durante o processo de encerramento do projeto, porém a captura das lições aprendidas deve ocorrer durante todo o ciclo de vida de projeto para garantir todas as informações sejam documentadas de forma estruturada.

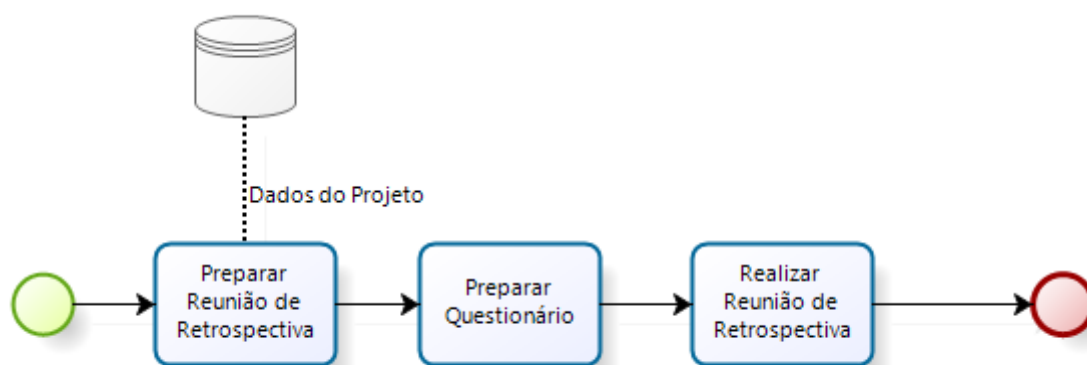


Figura 18 – Fase de Coleta.
Fonte: Produção da autora

Tabela 7 – Fase de Coleta

Nome	FASE DE COLETA
Objetivos	Nesta fase é realizado o levantamento das lições aprendidas. Os modelos analisados orientam a realização de uma revisão (retrospectiva) de pós-projeto (post-mortem) ou fase.
Atividades	<p>Preparar Reunião de Retrospectiva – Preparação das atividades da análise das lições aprendidas considerando o contexto do projeto, nessa atividade é reunida as informações quantitativas e qualitativas acerca do projeto em estudo. São tarefas desta atividade:</p> <ul style="list-style-type: none"> • Identificação e solicitação da documentação disponível. Devem ser analisados os registros históricos do projeto. • Definição do escopo da reunião de retrospectiva.

-
- Definição do time que irá participar da reunião de retrospectiva.

Preparar Questionário – O primeiro requisito para a concepção de um questionário é determinar quais questões devem ser feitas. Deve ser elaborado um questionário simples, direto e preciso. As questões mais importantes do projeto que determinam o sucesso ou fracasso do projeto devem ser o foco das questões, as quais devem ser distribuídas em categorias.

Realizar Reunião de Retrospectiva – Nessa etapa são coletadas experiências através de uma reunião, onde os participantes dão *feedback* direto sobre os eventos do projeto. Durante a reunião da equipe do projeto deve-se discutir as estratégias que contribuíram para o sucesso, bem como áreas de potencial melhoria.

Papeis e**Responsabilidades****Analista:**

- Responsável por preparar e monitorar reunião de retrospectiva;
- Responsável por elaborar e atualizar questionário;

Equipe de Projeto:

- Responsáveis por participar da reunião de retrospectiva e emitir opinião especializada.

Entradas	Informações históricas do projeto exemplos. Opinião especializada.
Saídas	Questionário genérico respondido. Levantamento dos sucessos e fracassos do projeto.

3.3.2 Processamento

Nesta fase devem-se documentar as lições aprendidas através do desenvolvimento de um relatório do pós-projeto. Pode-se adicionar valor às informações coletadas, tornando-as conhecimento.

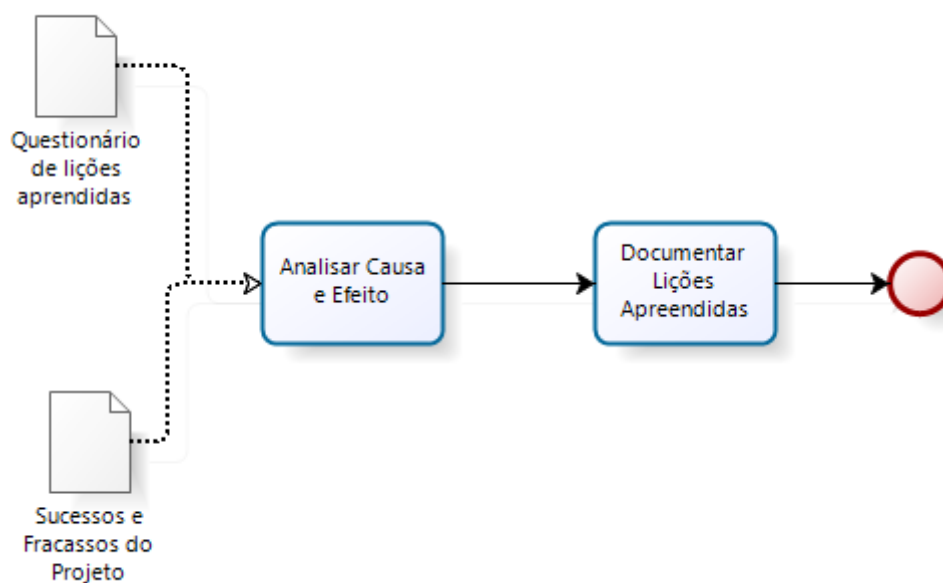


Figura 19 – Fase de Processamento
Fonte: Produção da autora

Tabela 8 - Fase de Processamento

Nome	FASE DE PROCESSAMENTO
-------------	------------------------------

Objetivos	Analisar os dados e as informações reunidas para avaliar as práticas atuais, determinar os problemas, os resultados recordes, e fazer recomendações para melhorias futuras do projeto.
Atividades	<p>Analisar Causa e Efeito– A Análise de Causa e Efeito é uma técnica para identificar as causas possíveis associadas a um determinado problema/efeito antes de se chegar a um número reduzido de causas principais que precisem ser tratadas.</p> <p>Documentar Lições Aprendidas– Trata-se de consolidar os resultados em tópicos. Além da categorização e descrição da lição, é importante afirmar descrever o impacto e fornecer uma recomendação para os gerentes de projeto considerarem em futuros projetos.</p>
Papeis e Responsabilidades	<p>Analista:</p> <ul style="list-style-type: none"> - Responsavel conduzir analise de causa e efeito; - Responsavel por elaborar relatório; <p>Equipe de Projeto:</p> <ul style="list-style-type: none"> - Responsaveis por apoiar analise de causa e efeito atraves de opinião especializada.
Entradas	<p>Questionário genérico respondido.</p> <p>Levantamento dos sucessos e fracassos do projeto.</p> <p>Opinião especializada.</p>
Saídas	Relatório de Lições Aprendidas

3.3.3 Armazenagem

Consiste em facilitar a localização, disponibilidade e utilização do conhecimento.

Os dados obtidos durante a fase de processamento podem ser armazenados em uma base de conhecimento.

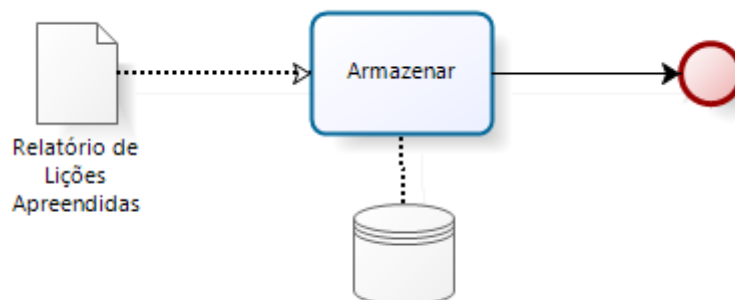


Figura 20 – Fase de Armazenamento
Fonte: Produção da autora

Tabela 9 – Fase de Armazenamento

Nome	FASE DE ARMAZENAMENTO
Objetivos	Consolidar a experiência adquirida na forma de nova ou atualizada e detalhada, modelos e outras formas de conhecimento estruturado adquiridos com projetos anteriores ou durante o projeto atual, e armazená-lo em uma base de experiência para que esteja disponível para projetos futuros.
Atividades	Armazenar – Armazenar o relatório de Lições aprendidas, bem como, organizar a documentação do projeto em uma base de dados.
Papeis e	Analista:

Responsabilidades	- Responsavel por armazenar as lições aprendidas.
--------------------------	---

Entradas	Relatório de Lições Aprendidas.
-----------------	---------------------------------

Saídas	Atualização dos ativos organizacionais.
---------------	---

3.3.4 Distribuição

A reutilização de experiência na forma de processos, produtos e outras formas de conhecimento são essenciais para melhoria da qualidade em uma organização, desta maneira não é suficiente apenas armazenar as lições, deve-se divulgar por toda a organização. Mas essa divulgação deve levar em consideração o direcionamento e a priorização destas informações de acordo com os interesses de cada grupo.

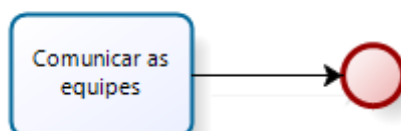


Figura 21 – Fase de distribuição
Fonte: Produção da autora

Tabela 10 –Fase de Distribuição

Nome	FASE DE DISTRIBUIÇÃO
Objetivos	Tornar disponíveis as lições aprendidas às pessoas da organização conforme apropriado.

Atividades	Comunicar as equipes – Trata-se de direcionar um comunicado as equipes da organização informando onde está disponível as lições aprendidas.
Papeis e Responsabilidades	Analista: - Responsavel por comunicar o resultado da analise de lições aprendidas.
Entradas	Ativos organizacionais.
Saídas	Comunicado as equipes

A figura abaixo mostra o resultado do fluxo do processo proposto.

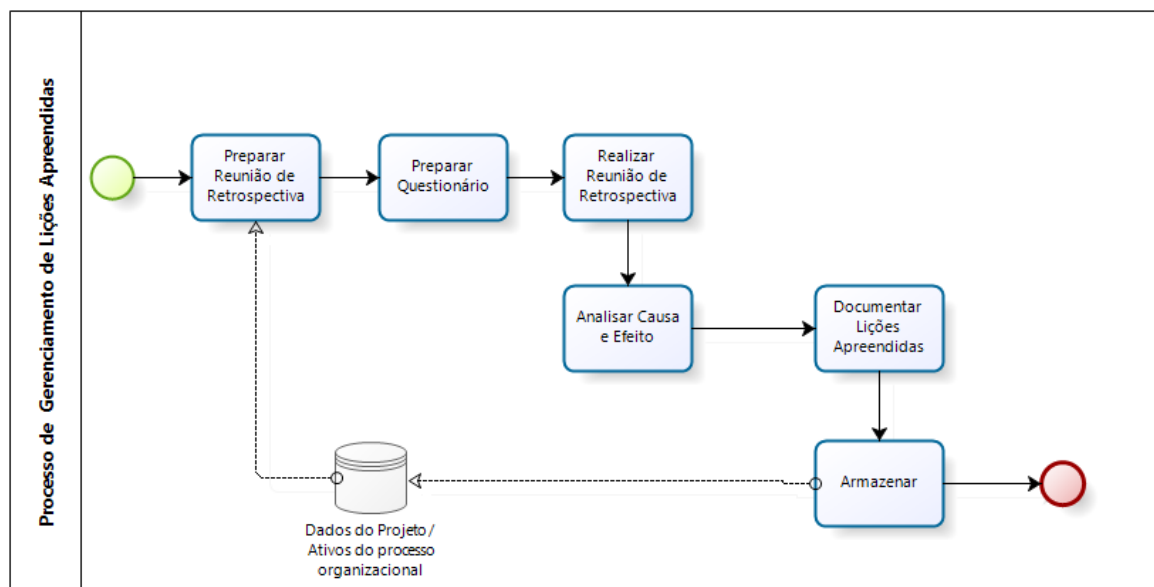


Figura 22 - Fluxo do processo proposto
Fonte: Produção da autora

No próximo capítulo será realizado um estudo de caso aplicando o processo definido em uma organização de software.

CAPÍTULO 4

ESTUDO DE CASO

Neste capítulo será planejado e executado um estudo de caso, por meio do conhecimento adquirido no embasamento teórico e análise do estado da arte.

4.1 Metodologia do Estudo de Caso

O estudo de caso aplicado neste trabalho levou em consideração a metodologia proposta por GIL (2002), a qual define um conjunto de etapas que podem ser seguidas na maioria das pesquisas definidas como estudos de caso:

- **Formulação do Problema:** foi formulado problema a ser resolvido durante os capítulos anteriores deste trabalho (fundamentação teórica e estado da arte).
- **Definição da Unidade-Caso:** os casos podem ser definidos do ponto de vista espacial ou temporal. Neste trabalho, a unidade caso é um projeto em uma organização de desenvolvimento de software. Caracteriza-se ainda como estudo de caso instrumental, que é aquele desenvolvido com o propósito de auxiliar no conhecimento ou redefinição de determinado problema.
- **Determinação do Número de Casos:** será constituído de um único caso.
- **Coleta de Dados:** servirá para conferir a significância do estudo de caso a seus resultados.

- **Avaliação e Análise dos Dados:** Será verificada a adequação do processo aos modelos, serão apresentadas as não conformidades e apontadas as melhorias para alcançar esta adequação.
- **Redação do Relatório:** o relatório referente ao estudo de caso constituído de um único caso foi elaborado sob a forma tradicional, através de uma narrativa. A redação do relatório segue nas próximas seções.

4.2 Organização

A organização pesquisada, na qual o estudo de caso foi aplicado, é a área de Engenharia de Software e a Fábrica de Software, ambas pertencentes à Diretoria de Tecnologia da empresa catarinense Nexxera.

Fundada em 1992, em Florianópolis (SC), a Nexxera, empresa de tecnologia de informação especializada em processamento de transações oferece ao mercado soluções de gestão de informação além de transmissão de dados críticos. Possui um portfólio de portais de serviço que integram e gerenciam as relações de seus clientes com os seus parceiros de negócios (clientes, fornecedores e bancos) através do conceito SaaS (Software as a service), com atuação nas áreas mercantil, logística e financeira.

4.3 Projeto Alvo

O projeto selecionado para avaliação é “Publicação de Nota Fiscal de Serviço Eletrônica (NFS-e)” no sistema Portal Financeiro.

O Portal Financeiro é um sistema em operação desde 2007, possui o objetivo de integrar as informações financeiras, através do gerenciamento de contas a

receber e contas a pagar de empresas de grande porte. Boa parte dos clientes desta solução SaaS são empresas que publicam compromissos e recebíveis para os seus respectivos clientes e fornecedores através do portal.

A publicação de NFS-e é um projeto iniciado em dezembro de 2011, e a finalização ocorreu em abril de 2012. O escopo do projeto consistiu na importação de NFS-e no padrão de cinco receitas municipais e apresentação das mesmas na WEB. A NFS-e é um documento fiscal emitido pelas prefeituras municipais do Brasil para recolhimento do imposto sobre serviço de qualquer natureza - ISS, esse documento é trafegado pela VAN Nexxera e seus dados são importados no Portal Financeiro, os usuários clientes da empresa contratante do serviço conseguem visualizar, pesquisar, emitir boletos através da solução.

A demanda deste projeto surgiu com a necessidade do departamento financeiro da própria Nexxera, com a publicação das NFS-e do município de Florianópolis. Foi realizada uma reunião inicial com o cliente e nela foram definidos os objetivos do projeto e seu escopo, como resultado foi gerado um documento inicial de requisitos e projeto. Esse documento foi orçado e aprovado pelo cliente, após esse documento foi detalhado através do refinamento dos requisitos, foi realizada uma reunião com desenvolvedores e analistas e iniciou-se o desenvolvimento dos requisitos especificados. A equipe de desenvolvimento foi dividida em duas: uma para tratar as informações trafegadas e outra para desenvolver a parte web, os testes são realizados de forma automatizada pelos desenvolvedores. Ao finalizar o desenvolvimento o sistema foi versionado com o novo módulo e a versão foi homologada pela analista de produto e, em seguida, entregue em produção. Por fim, foi realizado o gerenciamento de lições aprendidas através do processo modelado neste trabalho.

4.4 Relatório do Estudo de Caso

A equipe do projeto foi composta por seis pessoas, cujos papéis estão apresentados na Tabela 11. Participaram da reunião de retrospectiva três membros, pois os demais participantes encontram-se alocados em outra cidade, impossibilitando a participação.

Tabela 11 - Papel/Responsabilidade dos Membros da equipe

Projeto		Lições Aprendidas	
Papel / Responsabilidade	Quantidade Projeto	Quantidade Retrospectiva	Papel / Responsabilidade
Analista de produto e Projeto	1	1	moderador
Analista de Sistemas / Programador	2	1	suporte
Programador	3	1	suporte

Fase 1 - Coleta: A baixo a descrição de como foi realizada a aplicação da etapa 1 do processo proposto.

Atividade 1.1 - Preparar Reunião de Retrospectiva: Foram analisados os documentos referentes ao projeto disponíveis nos sistemas internos de gestão e repositórios do projeto. Os documentos analisados, os que mais provinham informações a respeito do projeto foram: documentos de proposta, detalhamento de requisitos, e resultado da homologação. Como resultado desta análise foi definido o escopo da reunião de retrospectiva, em seguida, foram convidados todos os integrantes do projeto via e-mail, apenas os que tinham disponibilidade de participar da reunião aceitaram o convite. Está atividade levou em torno de 1 hora e 30 min para leitura dos materiais do projeto e convocação para a reunião.

Atividade 1.2 - Preparar Questionário: Foi criado um relatório utilizando se como base o template proposto no apêndice A. As questões foram disponibilizadas no mecanismo de coleta de dados Google docs para formulários, o link do questionário foi encaminhado para todos os participantes da equipe do projeto. O questionário foi respondido por todos os membros da equipe, foram realizadas 30 perguntas qualitativas com as seguintes possibilidades de respostas (Muito, Suficiente, Pouco e Nada) as graduações para verificar quantitativamente os resultados dos eventos questionados foram respectivamente (3, 2, 1 e 0). Foram 6 respostas por questões, o somatório de respostas com graduação acima de 12 foram considerada possíveis sucessos e somatório com graduação abaixo de 12 foram considerados possíveis fracassos, somatório 12 foi considerado evento neutro. Os maiores fatores de sucesso e fracasso encontram-se abaixo.

Tabela 12 – Resultado do Questionário

Pergunta	Resultado
TEMPO - Quão perto o calendário real ficou em relação do cronograma projeto inicial?	16
RH - Quão bem os executivos apoiaram este projeto?	13
INTEGRAÇÃO - Quão bem a equipe do projeto compreendeu as expectativas de seus papéis e responsabilidades específicas?	13
ESCOPO - O quanto claramente foram definidos os objetivos para o seu trabalho?	13
RH - Quanto ativa e significativa foi o envolvimento das partes no projeto?	11
RH - Como foi o cumprimento do papel de cada membro da equipe do Projeto?	11
INTEGRAÇÃO - Quão eficiente e eficaz foram realizadas reuniões da equipe do projeto?	11
INTEGRAÇÃO - Quão eficiente foi a interação / cooperação entre equipes foi?	10
INTEGRAÇÃO - Quanto você se sentiu adequadamente envolvido em decisões de projeto?	10
QUALIDADE - O quanto os testes automáticos garantiram a programação?	9

COMUNICAÇÃO - Quão útil e completo era o repositório de documentos do projeto?	8
COMUNICAÇÃO - Treinamento foi realizado com antecedência 2 e / ou informações fornecidas para permitir que as pessoas afetadas pelas mudanças pudessem ajustar-se?	7
COMUNICAÇÃO - Quão completa e em tempo útil foram os materiais/documentos que foram fornecidos para decidir proceder de uma fase do ciclo de vida do projeto para o próximo?	7
RISCOS - Qual a precisão foi o Plano de Gestão de Risco / Log?	6

As respostas deste questionário serviram como base para a reunião de retrospectiva.

Atividade 1.4 Realizar Reunião de Retrospectiva - Participaram da reunião de retrospectiva a analistas de produtos como moderadora da reunião, analista de sistemas e um desenvolvedor. Foram utilizados cartões amarelos os quais são conhecidos como *post-its*. A reunião foi conduzida da seguinte maneira:

Nos cinco minutos iniciais da reunião foram apresentados os conceitos e objetivos da coleta de lições aprendidas, bem como, apresentados os resultados do questionário anteriormente aplicado; Em seguida, cada um dos membros escreveu nos *post-its* as ocorrências durante o gerenciamento e/ou o desenvolvimento do projeto, no mínimo, cinco cartões com ocorrências positivas e cinco cartões com ocorrências negativas. Nos cartões para ocorrências positivas foi assinalado “+” para eventos negativos foi assinalado “-”; Os cartões foram colados na parede e agrupados por categorias. As ocorrências citadas e categorizadas estão apresentadas na Tabela 10.

Tabela 13 - Tabela de ocorrências discutidas

Status	Ocorrências	Categorias
Negativa	Orçamentos superestimados	TEMPO
Positiva	Tempo o suficiente	
Positiva	Entrega na data e tempo o suficiente	
Positiva	Horas extras bem distribuídas	
Positiva	Sem atrasos	
Negativa	Manutenções no sistema durante o desenvolvimento	
Negativa	Documentação do sistema insuficiente	RISCOS
Negativa	Layout de arquivo tipo um "Frankstein"	RH
Positiva	Papeis e responsabilidades definidos	
Positiva	Capacidade técnica.	QUALIDADE
Negativa	Cliente Realizou Piloto em produção.	
Negativa	Falta de documentação do legado.	
Negativa	Falta de documento tecnico do legado.	
Negativa	Desconhecimento do layout	
Negativa	Sistema legado complexo	
Positiva	Aumento de conhecimento do sistema legado	
Positiva	Melhoria do layout de arquivo de entrada	
Positiva	Reconhecimento das regras do sistema legado	INTEGRAÇÃO
Positiva	Processo de desenvolvimento definido	
Negativa	Faltou o papel do analista de negócio.	ESCOPO
Negativa	Cliente interno com poder de decisão nas regras do sistema.	
Positiva	Sistema pode ser ofertado a novos clientes.	
Positiva	Regras de negócio e processo conhecidos	
Positiva	Pouca mudança de requisitos durante desenvolvimento	COMUNICAÇÃO
Negativa	Comunicação entre as equipes	
Negativa	Suporte não foi avisado apropriadamente da entrada em produção.	
Positiva	Envolvimentos dos usuários	
Positiva	Comunicação com o cliente interno	AQUISIÇÕES
Positiva	Boa ferramenta para gestão do projeto	
Negativa	Baixa prioridade do projeto	-

As ocorrências duplicadas ou que algum membro da equipe não concordou foram descartadas.

Fase 2 - Processamento: Nesta fase foi realizada a documentação das lições aprendidas.

Atividade 2.1 Analisar Causa e Efeito – A análise de causa e efeito é uma técnica para identificar as causas possíveis associadas a um determinado problema/efeito antes de se chegar a um número reduzido de causas principais que precisem ser tratadas. Após a categorização das ocorrências positivas e negativas do projeto, foi eleita a ocorrência “Orçamentos superestimados” para ser analisada. Cada membro citou durante a reunião de retrospectiva algumas possíveis causas do efeito “Orçamento superestimado” utilizando a técnica *brainstorming*, em seguida foi realizado o diagrama de espinha de peixe ou de Ishikawa.

O diagrama foi desenhado primeiramente utilizando uma folha A4, lápis e borracha. Todos os participantes da reunião da retrospectiva concordaram com o resultado final do diagrama, o qual posteriormente foi diagramado utilizando a ferramenta MS Visio e o resultado está apresentado na Figura 24.

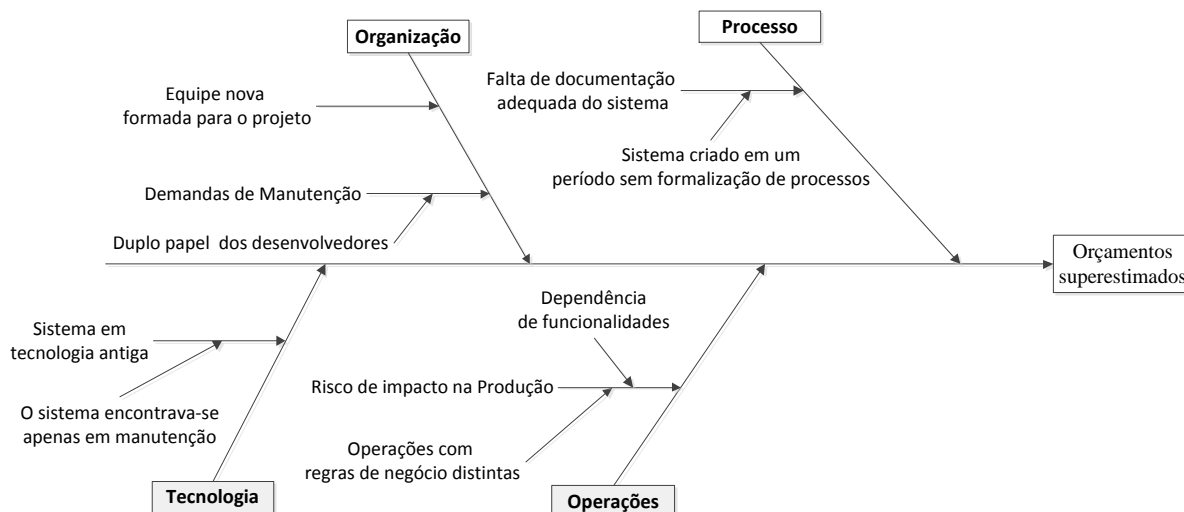


Figura 23 – Diagrama de Causa e Efeito

Para chegar ao resultado apresentado na figura 24 foram reunidas as listas iniciais de categorias principais e em seguida as causas detalhadas através da técnica de *Brainstorming* e, em seguida, iniciou-se a modelagem do diagrama.

Os membros da equipe da retrospectiva entraram em consenso que os principais fatos para o efeito “Orçamentos superestimados” foram a falta de documentação adequada do sistema, pois nem todos os membros conheciam o sistema antes do projeto; operações críticas em produção uma vez que, o sistema encontrava-se operante com funcionalidades complexas. Essas causas foram detalhadas por fim, o diagrama esclarece que o efeito do ponto negativo “orçamentos superestimados” possui como causa raiz o sistema estar em tecnologia antiga (antiga para os padrões da empresa, a qual sempre trabalha com tecnologia de ponta), porém em operação com regras de negócios distintas em funcionalidades compartilhadas por fim, a falta de documentação do legado possui como causa o sistema ter sido concebido em uma época em que os processos de documentação não eram formalizados.

Atividade 2.2 Documentar Lições Aprendidas – Foram consolidados os resultados em um documento único, apresentado no apêndice C deste trabalho. Foi criada a planilha de matriz de impacto e recomendação onde foi fornecida uma recomendação para projetos futuros.

Tabela 14 – Parte da documentação

Categoria Qualidade	Categoria Desenvolvimento	Ocorrência	Problema/ Sucesso	Impacto	Recomendação
Comunicação	Implantação	Suporte não foi avisado apropriadamente da entrada em produção.	Problema	A área não estava preparada para o número de ligações após a implantação.	Alem de treinar o suporte para as novas funcionalidade é importante formalizar a data de implantação para que a área esteja preparada para atender novos chamados de dúvidas.
Escopo	Requisitos	Cliente interno com poder de decisão nas regras do sistema.	Problema	Eventuais problemas durante a implementação.	O cliente deve apenas explicitar suas necessidades. A equipe do projeto propor a solução tecnicamente.
Integração	Modelagem de Negócio	Faltou o papel do analista de negócio.	Problema	Falta de conhecimento de detalhes sobre os processos do negocio envolvido, podem levar a não explicitação de requisitos na sua totalidade.	Envolver a área responsável pela análise de negócio sempre que possível.
Qualidade	Análise e Design	Desconhecimento do layout do sistema pela área responsável	Problema	Testes de qualidade com baixa aprovação, tempo de desenvolvimento e alteração longos.	Cada novo campo no layout do arquivo do sistema, deve-se documentar o motivo e a descrição em detalhes.
Tempo	Implantação	Manutenções no sistema durante o desenvolvimento	Problema	Duração excessivamente longa ou orçamentos superestimados	Sempre que possível separar a equipe em manutenção e desenvolvimento
Aquisições	Geral	Boa ferramenta para gestão do projeto	Sucesso	Boa rastreabilidade dos acontecimentos durante o projeto.	Registrar as ocorrências e atualizar as atividades delegadas no momento apropriado.
Escopo	Requisitos	Sistema pode ser ofertado a novos clientes.	Sucesso	Oferta de um produto pronto sem a necessidade de muitas customizações.	Sempre que possível conceber o sistema de uma maneira genérica que possa ser aproveitado em novos clientes com estrutura de negócio similar.
Qualidade	Geral	Aumento de conhecimento do sistema legado	Sucesso	Novos projetos possuíram uma documentação do legado.	Sempre que necessário realizar engenharia reversa para documentar o sistema. Caso o sistema seja legado e antigo, reserve horas para isso.
Qualidade	Análise e Design	Melhoria do layout de arquivo de entrada	Sucesso	Sempre que necessário a consulta do layout para obter alguma informação ele estará atualizado.	Sempre que possível realizar documentação de campos legados do layout.
Tempo	Projeto	Entrega na data e tempo suficiente	Sucesso	Tempo suficiente para realizar todas as etapas do processo de desenvolvimento	Sempre que possível colocar um % a mais no orçamento do desenvolvedor prevendo alguns riscos. Porém ficar atento para orçamentos superestimados.

Fase 3 - Armazenamento: Consolidar a experiência adquirida na forma de nova ou atualizada e detalhada.

Atividade 3.1 - Armazenar – O questionário realizado, os resultados coletados do questionário, os resultados da retrospectiva e a matriz de impacto e recomendação foram documentados e armazenados no repositório do projeto. Especificamente para a matriz de impacto e recomendação, foi criado um repositório comum a todos os projetos. Podendo somar-se a outras matrizes de análises de lições aprendidas, deste modo, será possível realizar uma consulta de riscos a serem monitorados em futuros projetos.

Fase 4 – Distribuição - Para tornar disponíveis as lições aprendidas.

Atividade 4.1 - Comunicar as equipes – Para informar a disponibilização das lições aprendidas aos interessados foi enviado um e-mail para toda a engenharia de

software. No corpo do e-mail foi explicado resumidamente o processo aplicado e onde ficou disponibilizadas as lições aprendidas com um link para tais informações.

4.5 Avaliação e Análise dos Dados

Nessa seção será verificada a adequação do processo aos modelos alvos. A classificação de conformidade do processo aos modelos PMI, CMMI, RUP e SCRUM está exposta na tabela abaixo.

Tabela 15 - Tabela de Avaliação do Processo

Classificação	Descrição
Totalmente (***)	Totalmente em consonância com os modelos alvos, sem fraquezas percebíveis.
Largamente (**)	Em consonância com os modelos alvos, com uma ou mais fraquezas percebíveis.
Parcialmente (*)	Implementa parcialmente algumas orientações dos modelos.
Não Implementado (-)	Não implementa nada do que foi orientado pelos modelos.

Analisando o resultado da avaliação apresentada na Tabela 12 é percebível que as atividades desenvolvidas para o processo proposto apresentam-se consonancia com os modelos estudados.

Tabela 16 – Avaliação do Processo

PMI	CMMI-DEV	RUP	SCRUM	Processo Genérico de Lições Aprendidas	Classificação do processo
1. Coleta					
Realizar uma revisão pós-projeto ou de final de fase.	Conduzir revisões periódicas da efetividade e da adequação do conjunto de processos-padrão da organização e de seus ativos de processo em relação aos objetivos estratégicos da organização.	Uma reunião post-mortem do projeto	Na Reunião de Retrospectiva (Retrospectiva) é realizada a revisão do processo de trabalho e onde é realizado o levantamento das lições aprendidas.	Roteiro para criação de questionário de lições aprendidas. preparação e realização da reunião para coleta de lições aprendidas.	***
2. Processamento					
Documentar as lições aprendidas.	Identificar lições aprendidas a partir da definição. Resultados de uma análise de causa e efeito.	Desenvolvimento de um relatório post-mortem do projeto.	-	Análise de causa e efeito e proposta de documentação em relatório.	***
3. Armazenagem					
Armazenamento no Ativos de Processos Organizacionais.	Atualização dos ativos de processo da organização.	-	-	Foi proposto o armazenar do relatório de Lições aprendidas em uma base comum da organização.	**
4. Distribuição					
-	Tornar disponíveis as lições aprendidas às pessoas da organização conforme apropriado.	-	-	Foi orientado um comunicado as equipes da organização informando onde está disponível as lições aprendidas.	**

CAPÍTULO 5

CONCLUSÕES

Na etapa final deste trabalho é apresentada a avaliação e conclusões do processo modelado através de um estudo de caso.

5.1 Considerações Finais

O principal resultado deste trabalho é um modelo de processo de gerenciamento de lições aprendidas em projetos de software. Esse modelo de processo unifica atividades dos frameworks de gestão (PMI), qualidade (CMMI), desenvolvimento (RUP) e desenvolvimento ágil (SCRUM). Foram resultados importantes também à revisão bibliográfica e a análise comparativa dos frameworks alvos.

O modelo de gerenciamento de lições aprendidas apresentou as seguintes características:

- O ciclo de vida do processo foi baseado na proposta de Angus e Patel para gestão do conhecimento, tal proposta compõem-se de quatro fases: coleta, processamento, armazenamento e distribuição;
- O processo desenvolvido possui descrições detalhadas de como realizar as atividades de cada fase do gerenciamento de lições aprendidas;
- As atividades o processo podem ser utilizadas de acordo com as fases propostas: ao final de um projeto ou durante a realização do mesmo;

- O processo pode ser aplicado em qualquer organização de software, independente do nível de maturidade no gerenciamento de processos;
- O modelo proposto utilizou como ferramentas planilhas, editores de textos, formulário web gratuito e etc. Não foi utilizada nenhuma ferramenta de apoio automatizada para a realização do processo.

Para o estudo de lições aprendidas e modelos de framework de melhorias de processo, qualidade e desenvolvimento foi utilizada uma vasta revisão bibliográfica baseada principalmente em artigos e guias consolidados que abordam o assunto, o resultado apresenta-se no capítulo 2. Acredita-se que este objetivo foi alcançado, pois através da leitura do capítulo pode-se compreender o contexto de lições aprendidas em projetos de softwares, através da apresentação dos modelos de referência eleitos. Em seguida, tomando-se como base a revisão bibliográfica, realizou-se uma análise comparativa dos frameworks de gestão (PMI), qualidade (CMMI), desenvolvimento (RUP) e desenvolvimento ágil (SCRUM), apresentando as suas principais características quanto ao gerenciamento de lições aprendidas. Com base na análise do estado da arte foi concebido o processo genérico abrangendo os modelos analisados. Com base no processo desenvolvido, foi planejado e executado um estudo de caso em uma organização de software. O estudo de caso apresentou a seguinte metodologia de formulação do problema, definição da unidade-caso, determinação do número de casos, coleta de dados. Ao final do estudo de caso foi realizada a avaliação do processo concebido para coleta de lições aprendidas o processo apresentou-se em conformidade aos modelos: PMI, CMMI, RUP e SCRUM. As fases do processo concebido obtiveram a classificação aderente ou totalmente aderente aos frameworks de melhores práticas.

O modelo desenvolvido foi avaliado em momento oportuno, ao fim de um projeto em uma organização de software. O estudo de caso realizado contribuiu para esse trabalho e para empresa onde o processo foi aplicado, pois com os resultados da fase de coleta foi possível levantar os principais problemas durante o projeto, seguido pela realização da análise das causas raízes. O resultado foi apresentado aos executivos da empresa juntamente com uma proposta de solução, a qual se encontra em análise.

5.2 Sugestão de Trabalhos Futuros

Como trabalhos futuros propõem-se os seguintes tópicos:

- Análise do processo de lições aprendidas para melhoria contínua de processos e/ou aprendizagem organizacional;
- Aprofundamento da revisão bibliográfica com foco na área de gestão do conhecimento;
- Concepção de uma ferramenta automatizada para apoiar o processo proposto;
- Extensão do processo para outras áreas de Gerenciamento de Projetos.

REFERÊNCIAS

ANGUS, J; PATEL, J. **Knowledge Management Cosmology**. *Information Week*, V. 673, March, 16/1998.

BASIL, V. ; CALDIERA, G., and ROMBACH. **Experience Factory**, Encyclopedia of Software Engineering Volume. 1:469-476, Marciniak, J. ed. John Wiley & Sons, 1994.

BASIL, V. ; CALDIERA, G. **Improve Software Quality by Reusing Knowledge and Experience**. Sloan Management Review, Fall 1995.

BECK, Kent. **Programação Extrema Explicada**. Porto Alegre: Bookman Companhia Editora, 2004.

BIRK, A. ; DINGSØYR, T. STALHANE, T. **Postmortem: Never Leave a Project without It**. IEEE Software, Junho, 2002.

CMMI Product Team. **CMMI for Development: Version 1.2**. Technical Report CMU/SEI-2006-TR-008, Carnegie Mellon University/Software Engineering Institute, August 2006.

CMMI Product Team. **CMMI for Development: Version 1.3**. Technical Report CMU/SEI-2010-TR-033, Carnegie Mellon University/Software Engineering Institute, Novembro, 2010.

DINGSØYR, T. ; MOE, N.B. ; NYTRØ. **Augmenting Experience Reports with Lightweight Postmortem Reviews**. Presented at Third International Conference on Product Focused Software Process Improvement, Kaiserslautern, Germany 2001.

FERENHOF, Helio A.; FORCELLINI, Fernando A. **Lições Aprendidas em um Projeto Mal Sucedido**. In: VI Congresso Brasileiro de Gerenciamento de Projetos do PMI, 2011, Fortaleza. Anais VI Congresso Brasileiro de Gerenciamento de Projetos do PMI, 2011.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GLASS, Robert L. **LOYAL OPPOSITION: Project Retrospectives, and Why They Never Happen**. IEEE Software, v.19 n.5, p.112, Setembro 2002.

IEEE, Computer Society. **SWEBOK: Guide to the Software Engineering Body of Knowledge**. California, IEEE, 2004.

JALOTE, Pankaj. **CMM in Practice: Processes for Executing Software Projects at Infosys**. Addison Wesley: Longman, 2000.

KERZNER, Harnold. **Gestão de Projetos: as Melhores Práticas**. Porto Alegre: Bookman.v. 2, 2007.

KERZNER, Harnold. **Project Management: A Systems Approach to Planning, Scheduling, and Controlling**. New York, NY. John Wiley & Sons 8th edn. ed. 2003.

KINOSHITA, Fumihiko. **Practices of an Agile Team**, Agile, 2008. AGILE '08. Conference.

KRUCHTEN, Philippe. **The Rational Unified Process: An Introduction**. Second Edition, Massachusetts: Addison-Wesley, 2001.

Manifesto Ágil. **Os Doze Princípios do Software Ágil**. <http://agilemanifesto.org/>, Visualizado em Outubro/2011.

MARTINS, José Carlos Cordeiro. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML**. Rio de Janeiro: Brasport Livros e Multimídia, 2006.

NONAKA, Ikujiro; TAKEUCHI, Hirotaka. **Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação**. Rio de Janeiro: Campus, 1997.

PAULK, Mark C.; CURTIS, William; Crissis, Mary B. & Weber, Charlie. **Capability Maturity Model for Software**. Version 1.1, Technical report CMU/SEI-93-TR-24, Software Engineering Institute, Pittsburgh, fevereiro, 1993.

PMI. **A Guide to the Project Management Body of Knowledge**. fourth ed., Project Management Institute (PMI), December 2008.

PROGRAMA BRASILEIRO DA QUALIDADE E PRODUTIVIDADE EM SOFTWARE. **Tecnologia da Informação, Programa Brasileiro da Qualidade e Produtividade em Software**. PBQP Software, 2009.

PROJECT MANAGEMENT INSTITUTE BRASIL. Inc. Benchmarking GP 2010. Rio de Janeiro. v. 7, 2010.

SARFRAZ, Farhan. **Managing for a Successful Project Closure**. CMET 09 2009 Portland International Conference on Management of Engineering Technology (pp. 1392-1395). IEEE. Islamabad, 2009.

SCHINDLER, Martin.; EPPLER, Martin. J. **Harvesting project knowledge**: a review of project learning methods and success factors. Journal of Project Management, v.21, p.219-228, 2003.

SCHWABER, Ken. **Scrum Guide**. Scrum Alliance, Maio, 2009.

SCHWABER, Ken. **Scrum Guide**. Scrum Alliance, Novembro, 2011.

SCHWABER, Ken. **Scrum Development Process**. OOPSLA'95 Workshop on Business Object Design and Implementation. Springer-Verlag. (1995).

APÊNDICES

APÊNDICE A - Questionário de lições aprendidas

Questionário de Lições Aprendidas

COMUNICAÇÃO - Quão bem as suas expectativas foram satisfeitas quanto à frequência e conteúdo da informação que foi transmitida a você pelo Gerente de Projeto?

- Muito
- Suficiente
- Pouco
- Nada

COMUNICAÇÃO - Quão completa e em tempo útil foram os materiais/documentos que foram fornecidos para decidir proceder de uma fase do ciclo de vida do projeto para o próximo?

- Muito
- Suficiente
- Pouco
- Nada

COMUNICAÇÃO - Quão eficientes foram conduzidas as reuniões da equipe do projeto?

- Muito
- Suficiente
- Pouco
- Nada

COMUNICAÇÃO - Quão eficaz os interessados do projeto foram comunicados?

- Muito
- Suficiente
- Pouco
- Nada

COMUNICAÇÃO - Quão útil e completo era o repositório de documentos do projeto?

- Muito
- Suficiente
- Pouco

- Nada

COMUNICAÇÃO - Treinamento foi realizado com antecedência suficiente e / ou informações fornecidas para permitir que as pessoas afetadas pelas mudanças pudessem ajustar-se?

- Muito
- Suficiente
- Pouco
- Nada

ESCOPO - Quanto claramente foram definidos os objetivos para este projeto?

- Muito
- Suficiente
- Pouco
- Nada

ESCOPO - O quanto claramente foram definidos os objetivos para o seu trabalho?

- Muito
- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quão efetivamente eram gerenciados questões sobre o projeto?

- Muito
- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quão efetivamente questões foram resolvidas sem afetar o cronograma do projeto ou orçamento?

- Muito
- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quão efetivamente as questões eram resolvidas sem a necessidade de escalar?

- Muito

- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quão útil foi o formato e o conteúdo do Relatório de Status do Projeto para você?

- Muito
- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quão eficiente foi a interação / cooperação entre equipes foi?

- Muito
- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quanto você se sentiu adequadamente envolvido em decisões de projeto?

- Muito
- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quão bem a equipe do projeto compreendeu as expectativas de seus papéis e responsabilidades específicas?

- Muito
- Suficiente
- Pouco
- Nada

INTEGRAÇÃO - Quão eficiente e eficaz foram realizadas reuniões da equipe do projeto?

- Muito
- Suficiente
- Pouco
- Nada

QUALIDADE - Quão bem definidos foram os critérios de aceitação de resultados do projeto?

- Muito

- Suficiente
- Pouco
- Nada

QUALIDADE - Quão perto do que foi definido dentro do escopo do projeto as entregas corresponderam?

- Muito
- Suficiente
- Pouco
- Nada

QUALIDADE - Quão bem foi os testes de unidade/homologação?

- Muito
- Suficiente
- Pouco
- Nada

QUALIDADE - Quão eficiente foi o design (ou implementação) das especificações?

- Muito
- Suficiente
- Pouco
- Nada

QUALIDADE - O quanto os testes automáticos garantiram a programação?

- Muito
- Suficiente
- Pouco
- Nada

RH - Quanto ativa e significativa foi o envolvimento das partes no projeto?

- Muito
- Suficiente
- Pouco
- Nada

RH - Quanto bem as suas expectativas foram satisfeitas quanto à extensão do seu envolvimento no projeto (esforço, os compromissos com o cronograma, etc)?

- Muito

- Suficiente
- Pouco
- Nada

RH - Como foi o cumprimento do papel de cada membro da equipe do Projeto?

- Muito
- Suficiente
- Pouco
- Nada

RH - Quão bem os executivos apoiaram este projeto?

- Muito
- Suficiente
- Pouco
- Nada

RH - Em geral, quão eficaz foi o desempenho do Gestor de Projeto?

- Muito
- Suficiente
- Pouco
- Nada

RH - Quanto claro foi o seu papel no projeto?

- Muito
- Suficiente
- Pouco
- Nada

RH - O quanto valorizado, reconhecido e recompensado você se sente pelos seus esforços?

- Muito
- Suficiente
- Pouco
- Nada

RISCOS - Qual a precisão foi o Plano de Gestão de Risco / Log?

- Muito
- Suficiente

- Pouco
- Nada

TEMPO - Quão perto o calendário real ficou em relação do cronograma projeto inicial?

- Muito
- Suficiente
- Pouco
- Nada

APÊNDICE B - Resultado do Qualitativo do Questionário

Pergunta	Membro 1	Membro 2	Membro 3	Membro 4	Membro 5	Membro 6
COMUNICAÇÃO - Quão bem as suas expectativas foram satisfeitas quanto à frequência e conteúdo da informação que foi transmitida a você pelo Gerente de Projeto?	Muito	Muito	Pouco	Pouco	Suficiente	Suficiente
COMUNICAÇÃO - Quão completa e em tempo útil foram os materiais/documentos que foram fornecidos para decidir proceder de uma fase do ciclo de vida do projeto para o próximo?	Nada	Pouco	Pouco	Suficiente	Pouco	Suficiente
COMUNICAÇÃO - Quão eficientes foram conduzidas as reuniões da equipe do projeto?	Suficiente	Suficiente	Pouco	Suficiente	Suficiente	Suficiente
COMUNICAÇÃO - Quão eficaz os interessados do projeto foram comunicados?	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
COMUNICAÇÃO - Quão útil e completo era o repositório de documentos do projeto?	Pouco	Suficiente	Pouco	Pouco	Pouco	Suficiente
COMUNICAÇÃO - Treinamento foi realizado com antecedência suficiente e / ou informações fornecidas para permitir que as pessoas afetadas pelas mudanças pudessem ajustar-se?	Suficiente	Suficiente	Suficiente	Suficiente	Pouco	Suficiente
ESCOPO - Quanto claramente foram definidos os objetivos para este projeto?	Pouco	Muito	Suficiente	Suficiente	Suficiente	Suficiente
ESCOPO - O quanto claramente foram definidos os objetivos para o seu trabalho?	Suficiente	Muito	Suficiente	Suficiente	Suficiente	Suficiente
INTEGRAÇÃO - Quão efetivamente eram gerenciados questões sobre o projeto?	Pouco	Muito	Suficiente	Suficiente	Suficiente	Suficiente
INTEGRAÇÃO - Quão efetivamente questões foram resolvidas sem afetar o cronograma do projeto ou orçamento?	Pouco	Muito	Suficiente	Suficiente	Suficiente	Suficiente
INTEGRAÇÃO - Quão efetivamente as questões eram resolvidas sem a necessidade de escalar?	Suficiente	Muito	Suficiente	Suficiente	Suficiente	Suficiente
INTEGRAÇÃO - Quão útil foi o formato e o conteúdo do Relatório de Status do Projeto para você?	Suficiente	Muito	Suficiente	Suficiente	Suficiente	Suficiente
INTEGRAÇÃO - Quão eficiente foi a interação / cooperação entre equipes foi?	Muito	Suficiente	Pouco	Pouco	Pouco	Suficiente

INTEGRAÇÃO - Quanto você se sentiu adequadamente envolvido em decisões de projeto?	Pouco	Suficiente	Suficiente	Pouco	Suficiente	Suficiente
INTEGRAÇÃO - Quão bem a equipe do projeto compreendeu as expectativas de seus papéis e responsabilidades específicas?	Muito	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
INTEGRAÇÃO - Quão eficiente e eficaz foram realizadas reuniões da equipe do projeto?	Muito	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
QUALIDADE - Quão bem definidos foram os critérios de aceitação de resultados do projeto?	Muito	Suficiente	Pouco	Suficiente	Suficiente	Suficiente
QUALIDADE - Quão perto do que foi definido dentro do escopo do projeto as entregas corresponderam?	Muito	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
QUALIDADE - Quão bem foi os testes de unidade/homologação?	Muito	Suficiente	Pouco	Muito	Suficiente	Suficiente
QUALIDADE - Quão eficiente foi o design (ou implementação) das especificações?	Suficiente	Suficiente	Pouco	Muito	Suficiente	Suficiente
QUALIDADE - O quanto os testes automáticos garantiram a programação?	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
RH - Quanto ativa e significativa foi o envolvimento das partes no projeto?	Suficiente	Suficiente	Pouco	Muito	Pouco	Suficiente
RH - Quanto bem as suas expectativas foram satisfeitas quanto à extensão do seu envolvimento no projeto (esforço, os compromissos com o cronograma, etc)?	Suficiente	Suficiente	Muito	Pouco	Suficiente	Suficiente
RH - Como foi o cumprimento do papel de cada membro da equipe do Projeto?	Suficiente	Suficiente	Pouco	Suficiente	Pouco	Suficiente
RH - Quão bem os executivos apoiaram este projeto?	Suficiente	Suficiente	Muito	Suficiente	Suficiente	Suficiente
RH - Em geral, quão eficaz foi o desempenho do Gestor de Projeto?	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
RH - Quanto claro foi o seu papel no projeto?	Suficiente	Muito	Suficiente	Suficiente	Suficiente	Suficiente
RH - O quanto valorizado, reconhecido e recompensado você se sente pelos seus esforços?	Nada	Suficiente	Suficiente	Pouco	Pouco	Suficiente
RISCOS - Qual a precisão foi o Plano de Gestão de Risco / Log?	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
TEMPO - Quão perto o calendário real ficou em relação do cronograma projeto inicial?	Suficiente	Suficiente	Muito	Pouco	Suficiente	Suficiente

PROCESSO UNIFICADO PARA GESTÃO DE LIÇÕES APRENDIDAS

Soraia Barbosa de Almeida

Centro Tecnológico - Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 - 88.040-970 - Florianópolis - SC – Brasil

soraiaa@gmail.com

Abstract. *This work consisted in address issues related to the process of management lessons learned in software projects. This will allow knowledge to be registered, stored and disseminated in a software organization. The model was proposed in order to be compatible with the main frameworks of best practices for TI control and management. A comparison was made between models with a focus on management lessons seized, then designed a process unifying the activities of each framework. To validate the proposed process, a study was conducted in a software organization. The process created is flexible and showed adherent to the models studied.*

Resumo. *Este trabalho aborda questões relacionadas ao processo de gerenciamento de lições aprendidas em projetos de software. Foi concebido um modelo de processo que permitirá que o conhecimento seja registrado, armazenado e disseminado em uma organização. O modelo proposto teve com o objetivo ser compatível com os principais frameworks de melhores práticas de controle e gerência de TI. Foi realizada uma comparação entre os modelos com foco na gestão de lições aprendidas e, em seguida, concebido um processo único para as atividades de cada framework. Para validar o processo proposto foi realizado um estudo de caso em uma organização de software. O processo criado mostrou-se flexível e aderente aos modelos estudados neste trabalho.*

1. Introdução. Para contribuir com o aprendizado em projetos, há necessidade de compartilhamento de conhecimento de forma a melhorar a relação entre os envolvidos, isso pode ocorrer através do gerenciamento de lições aprendidas. A documentação sistemática de percalços, erros ou armadilhas potenciais ajuda a reduzir os riscos do projeto (FERENHOF, 2011). A partir de uma perspectiva de longo prazo, um projeto de aprendizagem sistemática permite a empresa desenvolver competências de projeto que levam a uma vantagem competitiva sustentável (SCHINDLER, 2003).

Este trabalho possui como objetivo geral, a criação de um processo genérico para o gerenciamento de lições aprendidas aplicado em projetos, que permitirá que o conhecimento seja registrado, armazenado e disseminado em uma organização de software. Tal modelo pretende ser compatível com os principais frameworks de melhores práticas de controle e gerência de TI.

2. Modelos de Referência. A complexidade em arquiteturas de tecnologia da informação e infraestruturas é crescente. Por esse motivo, há muitos modelos de melhores práticas atualmente para a área da tecnologia da informação, os quais apoiam a implementação e melhoria dos processos. Desta maneira, foram eleitos para o desenvolvimento deste trabalho um framework para gestão, qualidade, desenvolvimento e desenvolvimento ágil. A seguir, serão apresentados cada um destes frameworks.

Tempo		1991	1994	1996	2001
Frameworks	Gestão			PMBOK PMI	
	Qualidade				CMMI SE
	Desenvolvimento		RUP Rational		
	Agil	SCRUM Schwaber & Sutherland			

Figura 24 – Frameworks e sua data de lançamento
Fonte: produção da própria autora

A análise de gestão de lições aprendidas tomou como base neste trabalho o conceito de Angus e Patel (1998), que define a gestão de conhecimento em quatro processos:

- Coleta : Consiste na introdução de informações e dados em um sistema.
- Processamento: Consiste em adicionar valor às informações coletadas, tornando-as conhecimento;
- Armazenagem: Consiste em facilitar a localização, disponibilidade e utilização do conhecimento.
- Distribuição: que consiste em levar conhecimento às pessoas que precisam usá-lo, pois seu valor decorre do uso.

Com base nesse modelo, a seguir, serão analisados os frameworks escopo deste trabalho. Analogamente os processos para gestão do conhecimento passaram a ser as fases para gestão de lições aprendidas neste trabalho.

3. Comparativo entre os modelos. Este tópico apresenta um mapeamento do gerenciamento de lições aprendidas orientadas pelos modelos: PMBOK, CMMI, RUP e SCRUM. O mapeamento objetiva investigar as similaridades e as divergências entre as abordagens desses modelos.

Lições Aprendidas no PMI - considera as lições aprendidas como: a aprendizagem obtida durante todo o processo de realização do projeto e/ou prestação de serviços. As lições aprendidas documentadas fazem parte dos “ativos de processos organizacionais” da empresa.

Para esse modelo, no mínimo, as lições aprendidas devem ser coletadas durante o encerramento do projeto, o guia orienta a realização de uma revisão de pós-projeto e a documentação das lições aprendidas, porém não é explícito como devem ser realizadas essas atividades. As descrições destas atividades fazem parte da fase de encerramento de projeto.

Lições Aprendidas no CMMI-DEV - as lições aprendidas fazem parte dos ativos de processo da organização. Através da área de processo de Análise e Resolução de Causas (CAR), as lições aprendidas são abordadas como o resultado de uma análise causa e efeito. Os defeitos e problemas podem ter sido previamente encontrados em outros projetos ou em fases ou tarefas iniciais de um projeto, as atividades de análise e resolução de causas podem ser vistas como um mecanismo de comunicação de lições aprendidas entre projetos. São citadas como um dos métodos para identificação de riscos, durante o processo Gestão de Riscos (RSKM).

Lições Aprendidas no RUP - apesar de sua versão padrão não tratar especificamente de oportunidades de aprendizado, no final de cada interação há um o momento para a realização formal das atividades de aprendizado. O RUP prevê o debate sobre lições aprendidas através da revisão de pós-projeto e ações corretivas no fechamento de cada iteração, ou seja, ocorre diversas vezes em um mesmo projeto.

Lições Aprendidas no SCRUM - na reunião de retrospectiva (retrospective) é realizada a revisão do processo de trabalho. É realizada o levantamento das lições aprendidas, essa reunião foca o framework e as práticas do Scrum, para se obter um melhor desempenho na próxima iteração. De acordo com Ken e Colin (2009), essa é uma reunião que permite a equipe do projeto discutir o sprint concluído de maneira a expressar o que poderia ser mudado, em relação a comunicação, o ambiente de trabalho, os artefatos e as ferramentas que são utilizadas durante o projeto. A reunião de retrospectiva objetiva a busca de uma melhor alternativa para tornar o próximo sprint mais agradável e produtivo.

Com base na análise realizada segue o comparativo dos modelos.

Tabela 17 – Comparativo dos modelos para Gerenciamento de Lições Aprendidas

PMI	CMMI-DEV	RUP	SCRUM
1. Coleta			

Realizar uma revisão pós-projeto ou de final de fase.	Conduzir revisões periódicas da efetividade e da adequação do conjunto de processos-padrão da organização e de seus ativos de processo em relação aos objetivos estratégicos da organização.	Uma reunião post-mortem do projeto	Na Reunião de Retrospectiva (Retrospective) é realizada a revisão do processo de trabalho. Realizado o levantamento das lições aprendidas.
2. Processamento			
Documentar as lições aprendidas.	Identificar lições aprendidas a partir da definição. Resultado de uma análise causa e efeito.	Desenvolvimento de um relatório post-mortem do projeto	-
3. Armazenagem			
Armazenamento no Ativos de Processos Organizacionais.	Atualização dos ativos de processo da organização.	-	-
4. Distribuição			
-	Tornar disponíveis as lições aprendidas às pessoas da organização conforme apropriado.	-	-

4. O processo. A seguir, serão descritas as fases do processo genérico criado com base na referencia bibliográfica e análise do estado da arte, utilizando a tabela de características proposta anteriormente.

Coleta: coletar lições aprendidas é uma parte integrante de todo projeto e serve a vários propósitos. A formalização das lições aprendidas pode ser realizada durante o processo de encerramento do projeto, porem a captura das lições aprendidas deve ocorrer durante todo o ciclo de vida de projeto para garantir todas as informações sejam documentadas de forma estruturada.

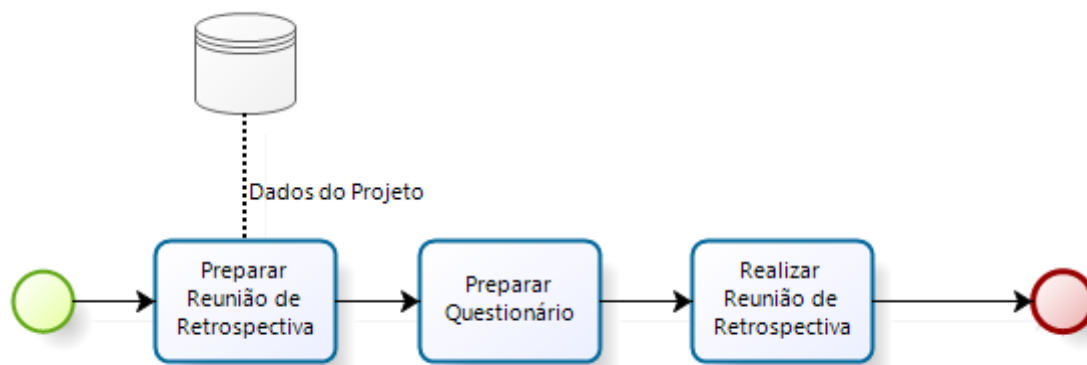


Figura 25 – Fase de Coleta.
Fonte: Produção da autora

Tabela 18 – Fase de Coleta

Nome	FASE DE COLETA
Objetivos	Nesta fase é realizado o levantamento das lições aprendidas. Os modelos analisados orientam a realização de uma revisão (retrospectiva) de pós-projeto (post-mortem) ou fase.
Atividades	<p>Preparar Reunião de Retrospectiva – Preparação das atividades da análise das lições aprendidas considerando o contexto do projeto, nessa atividade é reunida as informações quantitativas e qualitativas acerca do projeto em estudo. São tarefas desta atividade:</p> <ul style="list-style-type: none"> • Identificação e solicitação da documentação disponível. Devem ser analisados os registros históricos do projeto. • Definição do escopo da reunião de retrospectiva. • Definição do time que irá participar da reunião de retrospectiva. <p>Preparar Questionário – O primeiro requisito para a concepção de um questionário é determinar quais questões devem ser feitas. Deve ser elaborado um questionário simples, direto e preciso. As questões mais importantes do projeto que determinam o sucesso ou fracasso do projeto devem ser o foco das questões, as quais devem ser distribuídas em categorias.</p> <p>Realizar Reunião de Retrospectiva – Nessa etapa são</p>

coletadas experiências através de uma reunião, onde os participantes dão *feedback* direto sobre os eventos do projeto. Durante a reunião da equipe do projeto deve-se discutir as estratégias que contribuíram para o sucesso, bem como áreas de potencial melhoria.

Papeis e Responsabilidades	Analista: - Responsavel por preparar e monitorar reunião de retropectiva; - Responsavel por elaborar e atualizar questionário; Equipe de Projeto: - Responsaveis por participar da reunião de retropectiva e omitir opinião especializada.
Entradas	Informações históricas do projeto exemplos. Opinião especializada.
Saídas	Questionário genérico respondido. Levantamento dos sucessos e fracassos do projeto.

Processamento: Nesta fase devem-se documentar as lições aprendidas através do desenvolvimento de um relatório do pós-projeto. Pode-se adicionar valor às informações coletadas, tornando-as conhecimento.

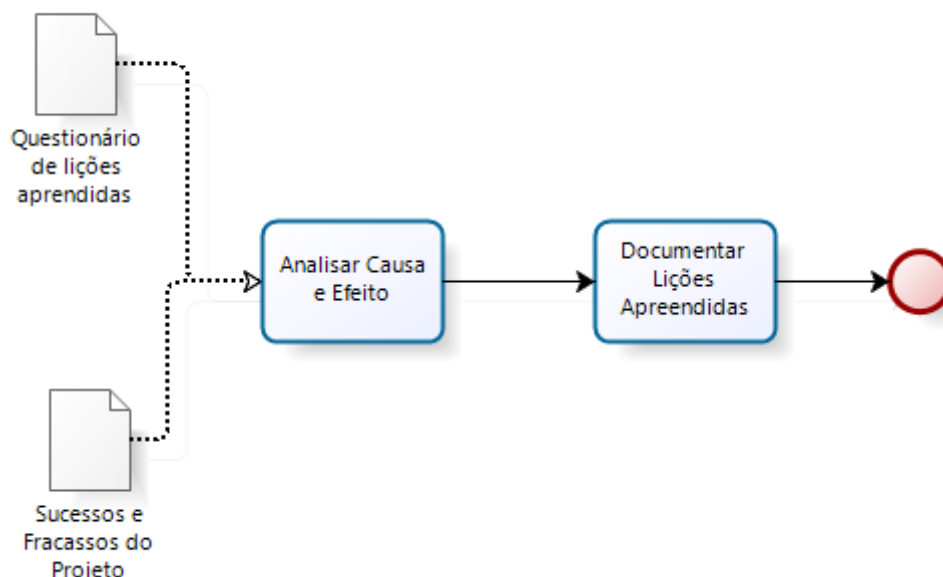


Figura 26 – Fase de Processamento
Fonte: Produção da autora

Tabela 19 - Fase de Processamento

Nome	FASE DE PROCESSAMENTO
Objetivos	Analisar os dados e as informações reunidas para avaliar as práticas atuais, determinar os problemas, os resultados recordes, e fazer recomendações para melhorias futuras do projeto.
Atividades	<p>Analisar Causa e Efeito– A Análise de Causa e Efeito é uma técnica para identificar as causas possíveis associadas a um determinado problema/efeito antes de se chegar a um número reduzido de causas principais que precisem ser tratadas.</p> <p>Documentar Lições Aprendidas– Trata-se de consolidar os resultados em tópicos. Além da categorização e descrição da lição, é importante afirmar descrever o impacto e fornecer uma recomendação para os gerentes de projeto considerarem em futuros projetos.</p>
Papeis e Responsabilidades	<p>Analista:</p> <ul style="list-style-type: none"> - Responsavel conduzir analise de causa e efeito; - Responsavel por elaborar relatório; <p>Equipe de Projeto:</p> <ul style="list-style-type: none"> - Responsaveis por apoiar analise de causa e efeito atraves de opinião especializada.
Entradas	<p>Questionário genérico respondido.</p> <p>Levantamento dos sucessos e fracassos do projeto.</p> <p>Opinião especializada.</p>
Saídas	Relatório de Lições Aprendidas

Armazenagem:

Consiste em facilitar a localização, disponibilidade e utilização do conhecimento.

Os dados obtidos durante a fase de processamento podem ser armazenados em uma base de conhecimento.

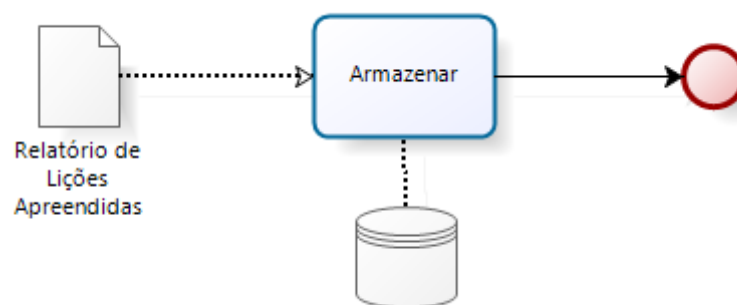


Figura 27 – Fase de Armazenamento
Fonte: Produção da autora

Tabela 20 – Fase de Armazenamento

Nome	FASE DE ARMAZENAMENTO
Objetivos	Consolidar a experiência adquirida na forma de nova ou atualizada e detalhada, modelos e outras formas de conhecimento estruturado adquiridos com projetos anteriores ou durante o projeto atual, e armazená-lo em uma base de experiência para que esteja disponível para projetos futuros.
Atividades	Armazenar – Armazenar o relatório de Lições aprendidas, bem como, organizar a documentação do projeto em uma base de dados.
Papeis e Responsabilidades	Analista: - Responsável por armazenar as lições aprendidas.
Entradas	Relatório de Lições Apreendidas.
Saídas	Atualização dos ativos organizacionais.

Distribuição:

A reutilização de experiência na forma de processos, produtos e outras formas de conhecimento são essenciais para melhoria da qualidade em uma organização, desta maneira não é suficiente apenas armazenar as lições, deve-se divulgar por toda a organização. Mas essa divulgação deve levar em consideração o direcionamento e a priorização destas informações de acordo com os interesses de cada grupo.

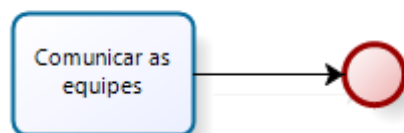


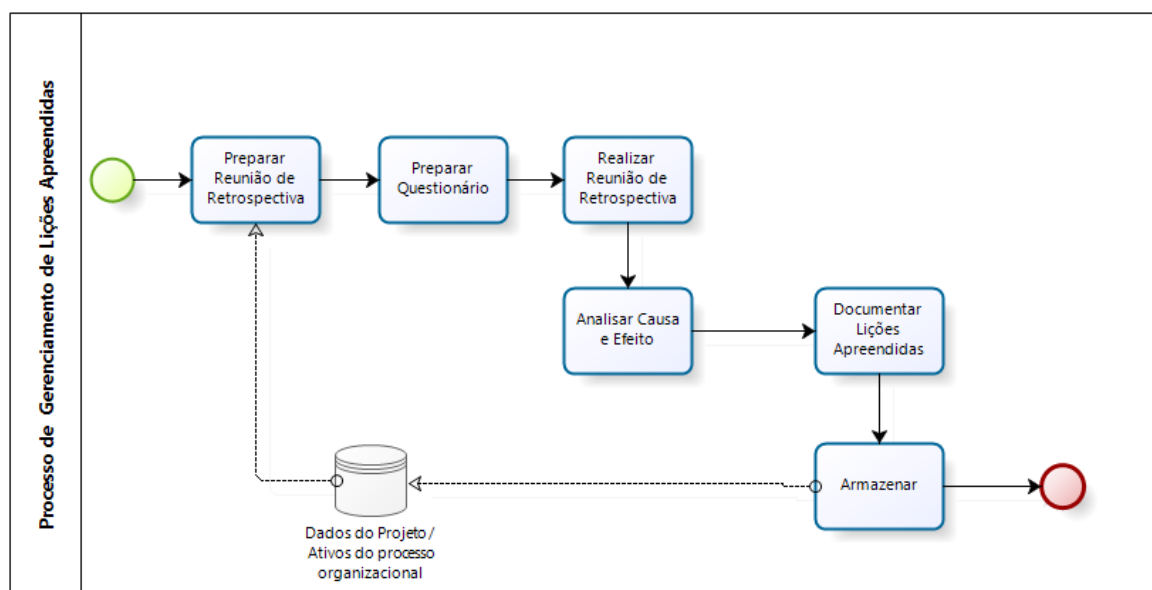
Figura 28 – Fase de distribuição

Fonte: Produção da autora

Tabela 21 – Fase de Distribuição

Nome	FASE DE DISTRIBUIÇÃO
Objetivos	Tornar disponíveis as lições aprendidas às pessoas da organização conforme apropriado.
Atividades	Comunicar as equipes – Trata-se de direcionar um comunicado as equipes da organização informando onde está disponível as lições aprendidas.
Papeis e Responsabilidades	Analista: - Responsável por comunicar o resultado da análise de lições aprendidas.
Entradas	Ativos organizacionais.
Saídas	Comunicado as equipes

A figura abaixo mostra o resultado do fluxo do processo proposto.



5. Estudo de Caso. O estudo de caso aplicado neste trabalho levou em consideração a metodologia proposta por GIL (2002), a qual define um conjunto de etapas que podem ser seguidas na maioria das pesquisas definidas como estudos de caso.

A organização pesquisada, na qual o estudo de caso foi aplicado, foi uma empresa de tecnologia catarinense. A equipe do projeto foi composta por seis pessoas, cujos papéis estão apresentados na Tabela 11. Participaram da reunião de retrospectiva três membros, pois os demais participantes encontram-se alocados em outra cidade, impossibilitando a participação.

Tabela 22 - Papel/Responsabilidade dos Membros da equipe

Projeto		Lições Aprendidas	
Papel / Responsabilidade	Quantidade Projeto	Quantidade Retrospectiva	Papel / Responsabilidade
Analista de produto e Projeto	1	1	moderador
Analista de Sistemas / Programador	2	1	suporte
Programador	3	1	suporte

Fase 1 - Coleta: A baixo a descrição de como foi realizada a aplicação da etapa 1 do processo proposto.

Atividade 1.1 - Preparar Reunião de Retrospectiva: Foram analisados os documentos referentes ao projeto disponíveis nos sistemas internos de gestão e repositórios do projeto. Os documentos analisados, os que mais provinham informações a respeito do projeto foram: documentos de proposta, detalhamento de requisitos, e resultado da homologação. Como resultado desta análise foi definido o escopo da reunião de retrospectiva, em seguida, foram convidados todos os integrantes do projeto via e-mail, apenas os que tinham disponibilidade de participar da reunião aceitaram o convite. Esta atividade levou em torno de 1 hora e 30 min para leitura dos materiais do projeto e convocação para a reunião.

Atividade 1.2 - Preparar Questionário: Foi criado um relatório utilizando se como base o template proposto no apêndice A. As questões foram disponibilizadas no mecanismo de coleta de dados Google docs para formulários, o link do questionário foi encaminhado para todos os participantes da equipe do projeto. O questionário foi respondido por todos os membros da equipe, foram realizadas 30 perguntas qualitativas com as seguintes possibilidades de respostas (Muito, Suficiente, Pouco e Nada) as graduações para verificar quantitativamente os resultados dos eventos questionados foram respectivamente (3, 2, 1 e 0). Foram 6 respostas por questões, o somatório de respostas com graduação acima de 12 foram considerada possíveis sucessos e somatório com graduação abaixo de 12

foram considerados possíveis fracassos, somatório 12 foi considerado evento neutro. Os maiores fatores de sucesso e fracasso encontram-se abaixo.

Atividade 1.4 Realizar Reunião de Retrospectiva - Participaram da reunião de retrospectiva a analistas de produtos como moderadora da reunião, analista de sistemas e um desenvolvedor. Foram utilizados cartões amarelos os quais são conhecidos como *post-its*. A reunião foi conduzida da seguinte maneira:

Nos cinco minutos iniciais da reunião foram apresentados os conceitos e objetivos da coleta de lições aprendidas, bem como, apresentados os resultados do questionário anteriormente aplicado; Em seguida, cada um dos membros escreveu nos *post-its* as ocorrências durante o gerenciamento e/ou o desenvolvimento do projeto, no mínimo, cinco cartões com ocorrências positivas e cinco cartões com ocorrências negativas. Nos cartões para ocorrências positivas foi assinalado “+” para eventos negativos foi assinalado “-”; Os cartões foram colados na parede e agrupados por categorias. As ocorrências citadas e categorizadas estão apresentadas na Tabela 10.

Fase 2 - Processamento: Nesta fase foi realizada a documentação das lições aprendidas.

Atividade 2.1 Analisar Causa e Efeito – A análise de causa e efeito é uma técnica para identificar as causas possíveis associadas a um determinado problema/efeito antes de se chegar a um número reduzido de causas principais que precisem ser tratadas. Após a categorização das ocorrências positivas e negativas do projeto, foi eleita a ocorrência “Orçamentos superestimados” para ser analisada. Cada membro citou durante a reunião de retrospectiva algumas possíveis causas do efeito “Orçamento superestimado” utilizando a técnica *brainstorming*, em seguida foi realizado o diagrama de espinha de peixe ou de Ishikawa.

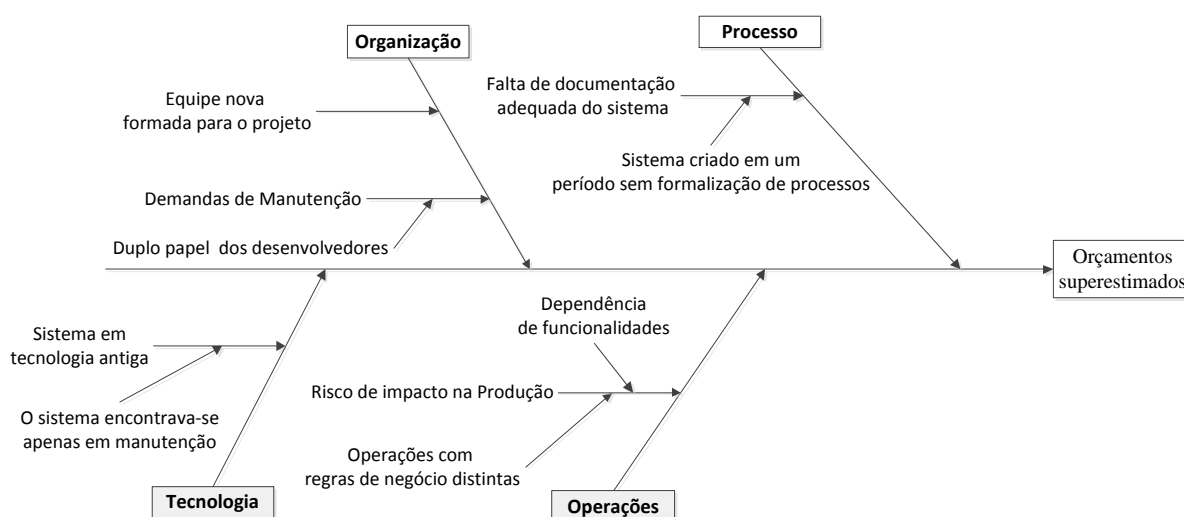


Figura 29 – Diagrama de Causa e Efeito

Atividade 2.2 Documentar Lições Aprendidas – Foram consolidados os resultados em um documento único, apresentado no apêndice C deste trabalho. Foi criada a planilha de matriz de impacto e recomendação onde foi fornecida uma recomendação para projetos futuros.

Fase 3 - Armazenamento: Consolidar a experiência adquirida na forma de nova ou atualizada e detalhada.

Atividade 3.1 - Armazenar – O questionário realizado, os resultados coletados do questionário, os resultados da retrospectiva e a matriz de impacto e recomendação foram documentados e armazenados no repositório do projeto. Especificamente para a matriz de impacto e recomendação, foi criado um repositório comum a todos os projetos. Podendo somar-se a outras matrizes de análises de lições aprendidas, deste modo, será possível realizar uma consulta de riscos a serem monitorados em futuros projetos.

Fase 4 – Distribuição - Para tornar disponíveis as lições aprendidas.

Atividade 4.1 - Comunicar as equipes – Para informar a disponibilização das lições aprendidas aos interessados foi enviado um e-mail para toda a engenharia de software. No corpo do e-mail foi explicado resumidamente o processo aplicado e onde ficou disponibilizadas as lições aprendidas com um link para tais informações.

6. Conclusão.

O modelo de gerenciamento de lições aprendidas apresentou as seguintes características:

- O ciclo de vida do processo foi baseado na proposta de Angus e Patel para gestão do conhecimento, tal proposta compõem-se de quatro fases: coleta, processamento, armazenamento e distribuição;
- O processo desenvolvido possui descrições detalhadas de como realizar as atividades de cada fase do gerenciamento de lições aprendidas;
- As atividades o processo podem ser utilizadas de acordo com as fases propostas: ao final de um projeto ou durante a realização do mesmo;
- O processo pode ser aplicado em qualquer organização de software, independente do nível de maturidade no gerenciamento de processos;
- O modelo proposto utilizou como ferramentas planilhas, editores de textos, formulário web gratuito e etc. Não foi utilizada nenhuma ferramenta de apoio automatizada para a realização do processo.

Para o estudo de lições aprendidas e modelos de framework de melhorias de processo, qualidade e desenvolvimento foi utilizada uma vasta revisão bibliográfica baseada principalmente em artigos e guias consolidados que abordam o assunto, o resultado apresenta-se no capítulo 2. Acredita-se que este objetivo foi alcançado,

pois através da leitura do capítulo pode-se compreender o contexto de lições aprendidas em projetos de softwares, através da apresentação dos modelos de referência eleitos. Em seguida, tomando-se como base a revisão bibliográfica, realizou-se uma análise comparativa dos frameworks de gestão (PMI), qualidade (CMMI), desenvolvimento (RUP) e desenvolvimento ágil (SCRUM), apresentando as suas principais características quanto ao gerenciamento de lições aprendidas. Com base na análise do estado da arte foi concebido o processo genérico abrangendo os modelos analisados. Com base no processo desenvolvido, foi planejado e executado um estudo de caso em uma organização de software. O estudo de caso apresentou a seguinte metodologia de formulação do problema, definição da unidade-caso, determinação do número de casos, coleta de dados. Ao final do estudo de caso foi realizada a avaliação do processo concebido para coleta de lições aprendidas o processo apresentou-se em conformidade aos modelos: PMI, CMMI, RUP e SCRUM. As fases do processo concebido obtiveram a classificação aderente ou totalmente aderente aos frameworks de melhores práticas.

O modelo desenvolvido foi avaliado em momento oportuno, ao fim de um projeto em uma organização de software. O estudo de caso realizado contribuiu para esse trabalho e para empresa onde o processo foi aplicado, pois com os resultados da fase de coleta foi possível levantar os principais problemas durante o projeto, seguido pela realização da análise das causas raízes. O resultado foi apresentado aos executivos da empresa juntamente com uma proposta de solução, a qual se encontra em análise.

4.1 Sugestão de Trabalhos Futuros

Como trabalhos futuros propõem-se os seguintes tópicos:

- Análise do processo de lições aprendidas para melhoria contínua de processos e/ou aprendizagem organizacional;
- Aprofundamento da revisão bibliográfica com foco na área de gestão do conhecimento;
- Concepção de uma ferramenta automatizada para apoiar o processo proposto;
- Extensão do processo para outras áreas de Gerenciamento de Projetos.

REFERÊNCIAS

ANGUS, J; PATEL, J. **Knowledge Management Cosmology**. *Information Week*, V. 673, March,16/1998.

BASILI, V. ; CALDIERA, G., and ROMBACH. **Experience Factory**, Encyclopedia of Software Engineering Volume. 1:469-476, Marciniak, J. ed. John Wiley & Sons, 1994.

BASILI, V. ; CALDIERA, G. **Improve Software Quality by Reusing Knowledge and Experience**. Sloan Management Review, Fall 1995.

BECK, Kent. **Programação Extrema Explicada**. Porto Alegre: Bookman Companhia Editora, 2004.

BIRK, A. ; DINGSØYR, T. STALHANE, T. **Postmortem**: Never Leave a Project without It. IEEE Software, Junho, 2002.

CMMI Product Team. **CMMI for Development**: Version 1.2. Technical Report CMU/SEI-2006-TR-008, Carnegie Mellon University/Software Engineering Institute, August 2006.

CMMI Product Team. **CMMI for Development**: Version 1.3. Technical Report CMU/SEI-2010-TR-033, Carnegie Mellon University/Software Engineering Institute, Novembro, 2010.

DINGSØYR, T. ; MOE, N.B. ; NYTRØ. **Augmenting Experience Reports with Lightweight Postmortem Reviews**. Presented at Third International Conference on Product Focused Software Process Improvement, Kaiserslautern, Germany 2001.

FERENHOF, Helio A.; FORCELLINI, Fernando A. **Lições Aprendidas em um Projeto Mal Sucedido**. In: VI Congresso Brasileiro de Gerenciamento de Projetos do PMI, 2011, Fortaleza. Anais VI Congresso Brasileiro de Gerenciamento de Projetos do PMI, 2011.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GLASS, Robert L. **LOYAL OPPOSITION: Project Retrospectives, and Why They Never Happen**. IEEE Software, v.19 n.5, p.112, Setembro 2002.

IEEE, Computer Society. **SWEBOK: Guide to the Software Engineering Body of Knowledge**. California, IEEE, 2004.

JALOTE, Pankaj. **CMM in Practice: Processes for Executing Software Projects at Infosys**. Addison Wesley: Longman, 2000.

KERZNER, Harnold. **Gestão de Projetos: as Melhores Práticas**. Porto Alegre: Bookman.v. 2, 2007.

KERZNER, Harnold. **Project Management: A Systems Approach to Planning, Scheduling, and Controlling**. New York, NY. John Wiley & Sons 8th edn. ed. 2003.

KINOSHITA, Fumihiko. **Practices of an Agile Team**, Agile, 2008. AGILE '08. Conference.

KRUCHTEN, Philippe. **The Rational Unified Process: An Introduction**. Second Edition, Massachusetts: Addison-Wesley, 2001.

Manifesto Ágil. **Os Doze Princípios do Software Ágil**. <http://agilemanifesto.org/>, Visualizado em Outubro/2011.

MARTINS, José Carlos Cordeiro. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML**. Rio de Janeiro: Brasport Livros e Multimídia, 2006.

NONAKA, Ikujiro; TAKEUCHI, Hirotaka. **Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação**. Rio de Janeiro: Campus, 1997.

PAULK, Mark C.; CURTIS, William; Crissis, Mary B. & Weber, Charlie. **Capability Maturity Model for Software**. Version 1.1, Technical report CMU/SEI-93-TR-24, Software Engineering Institute, Pittsburgh, fevereiro, 1993.

PMI. **A Guide to the Project Management Body of Knowledge**. fourth ed., Project Management Institute (PMI), December 2008.

PROGRAMA BRASILEIRO DA QUALIDADE E PRODUTIVIDADE EM SOFTWARE. **Tecnologia da Informação, Programa Brasileiro da Qualidade e Produtividade em Software**. PBQP Software, 2009.

PROJECT MANAGEMENT INSTITUTE BRASIL. Inc. Benchmarking GP 2010. Rio de Janeiro. v. 7, 2010.

SARFRAZ, Farhan. **Managing for a Successful Project Closure**. CMET 09 2009 Portland International Conference on Management of Engineering Technology (pp. 1392-1395). IEEE. Islamabab, 2009.

SCHINDLER, Martin.; EPPLER, Martin. J. **Harvesting project knowledge**: a review of project learning methods and success factors. Journal of Project Management, v.21, p.219-228, 2003.

SCHWABER, Ken. **Scrum Guide**. Scrum Alliance, Maio, 2009.

SCHWABER, Ken. **Scrum Guide**. Scrum Alliance, Novembro, 2011.

SCHWABER, Ken. **Scrum Development Process**. OOPSLA'95 Workshop on Business Object Design and Implementation. Springer-Verlag. (1995).

