

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

**Uma Análise sobre a Gestão do Conhecimento no Processo
de Desenvolvimento de Software**

Mauricio Isac Cardoso

TCC-EP-82-2013

Maringá - Paraná
Brasil

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

**Uma Análise sobre a Gestão do Conhecimento no Processo
de Desenvolvimento de Software**

Mauricio Isac Cardoso

TCC-EP-82-2013

Trabalho de Conclusão de Curso apresentado como requisito de avaliação no curso de graduação em Engenharia de Produção na Universidade Estadual de Maringá – UEM.

Orientador(a): Prof.(^a): Msc. Gislaine Camila Lapasini Leal

**Maringá - Paraná
2013**

AGRADECIMENTOS

Agradeço inicialmente Deus, que me deu forças e energias para encarar todas as dificuldades, que não foram poucas.

Agradeço também a minha esposa, Renata de Carvalho Almeida Cardoso, pessoa que sempre esteve ao meu lado me apoiando, me corrigindo e caminhando sempre comigo, não importasse o momento que estávamos.

Muito obrigado a todos os professores do Departamento de Engenharia de Produção da UEM. Especialmente para a professora Gislaine Camila Lapasini Leal, que acreditou em meu trabalho e teve toda paciência para me ajudar e finalizá-lo,

Agradeço a minha família, por ter me apoiado nos momentos difíceis e por sempre ser o meu porto seguro, a minha estrutura básica. Não conseguiria sequer começar nenhum projeto se não fosse todos os investimentos que foram feitos por mim.

E por último, mas não menos importante, agradeço a todos meus amigos e colegas que participaram deste projeto. Em qualquer momento, sempre tive alguém para me ouvir, me aconselhar e me ensinar coisas novas.

RESUMO

A gestão do conhecimento se mostra uma prática extremamente eficiente para a retenção do conhecimento dentro das empresas. E isso se mostra crucial quando o produto da empresa depende totalmente desse conhecimento. As empresas de software precisam garantir que o conhecimento dos funcionários sejam desenvolvidos e retidos na organização.

Com a constante necessidade de trabalhar o conhecimento dentro das empresas de software, este trabalho tem como objetivo fazer uma análise teórica específica da gestão do conhecimento no processo de desenvolvimento de software, traçando um possível modelo de representação das conversões do conhecimento em cada processo de desenvolvimento de software. Também serão analisadas as práticas e ferramentas que possivelmente darão apoio para tais conversões do conhecimento.

Palavras-chave: Gestão do Conhecimento, Processo de desenvolvimento de software, práticas de gestão do conhecimento, ferramentas de gestão do conhecimento.

ABSTRACT

The knowledge management has shows itself as a efficient practice to retain knowledge inside the companies. This is crucial when the company product depends this knowledge totally. Software companies need to ensure that the employees knowledge are developed and retained inside the organization.

With the constantly need to work the knowledge inside software companies this paper has as objective bring a specific analysis of knowledge management in software development process, bringing a knowledge conversions representative model possible in each software development process. It will be analyzed tools and practices that will support those knowledge conversions.

Keywords: Knowledge Management, Software Development Process, knowledge management practices, knowledge management tools.

LISTA DE FIGURAS

FIGURA 1 - REPRESENTAÇÃO DO MODELO EM CASCATA	16
FIGURA 2 - MODELO SECI	23
FIGURA 3 - REPRESENTAÇÃO DO MODELO EM CASCATA	34
FIGURA 4 - CONVERSÕES DO CONHECIMENTO NA ANÁLISE DE REQUISITOS	37
FIGURA 5 - CONVERSÕES DO CONHECIMENTO NO PROJETO DE SISTEMAS.	42
FIGURA 6 - CONVERSÕES DO CONHECIMENTO NA IMPLEMENTAÇÃO.	46
FIGURA 7 - CONVERSÕES DO CONHECIMENTO NA INTEGRAÇÃO.	50
FIGURA 8 - CONVERSÕES DO CONHECIMENTO NA MANUTENÇÃO.	54
FIGURA 9 – PROPOSTA DE CONVERSÕES DO CONHECIMENTO EM COMUM NO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS.	58
FIGURA 10 – MODELO DE CONVERSÕES DO CONHECIMENTO EXCLUSIVAS DE CADA PROCESSO.	59

LISTA DE QUADROS

QUADRO 1 – DELIMITAÇÃO DE FERRAMENTAS DE TI COM OS DEFINIDOS GRUPOS DE FERRAMENTAS DE GESTÃO DO CONHECIMENTO.....	35
QUADRO 2 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A SOCIALIZAÇÃO NA ANÁLISE DE REQUISITOS	38
QUADRO 3 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A EXTERNALIZAÇÃO NA ANÁLISE DE REQUISITOS	39
QUADRO 4 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A COMBINAÇÃO NA ANÁLISE DE REQUISITOS	40
QUADRO 5 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A INTERNALIZAÇÃO NA ANÁLISE DE REQUISITOS	40
QUADRO 6 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A SOCIALIZAÇÃO NO PROJETO DE SISTEMAS.....	43
QUADRO 7 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A EXTERNALIZAÇÃO NO PROJETO DE SISTEMAS.....	43
QUADRO 8 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A COMBINAÇÃO NO PROJETO DE SISTEMAS.....	44
QUADRO 9 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A INTERNALIZAÇÃO NO PROJETO DE SISTEMAS.....	45
QUADRO 10 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A SOCIALIZAÇÃO NA IMPLEMENTAÇÃO.....	47
QUADRO 11 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A EXTERNALIZAÇÃO NA IMPLEMENTAÇÃO.....	48
QUADRO 12 – FERRAMENTAS RECOMENDADAS PARA A COMBINAÇÃO NA IMPLEMENTAÇÃO.....	48
QUADRO 13 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A INTERNALIZAÇÃO NA IMPLEMENTAÇÃO.....	49
QUADRO 14 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A SOCIALIZAÇÃO NA INTEGRAÇÃO.....	51
QUADRO 15 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A EXTERNALIZAÇÃO NA INTEGRAÇÃO.....	52
QUADRO 16 – FERRAMENTAS RECOMENDADAS PARA A COMBINAÇÃO NA INTEGRAÇÃO..	52
QUADRO 17 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A INTERNALIZAÇÃO NA INTEGRAÇÃO.....	53
QUADRO 18 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A SOCIALIZAÇÃO NA MANUTENÇÃO.....	55
QUADRO 19 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A EXTERNALIZAÇÃO NA INTEGRAÇÃO.....	56
QUADRO 20 – FERRAMENTAS RECOMENDADAS PARA A COMBINAÇÃO NA MANUTENÇÃO	56
QUADRO 21 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A INTERNALIZAÇÃO NA MANUTENÇÃO.....	57
QUADRO 22 – PRÁTICAS E FERRAMENTAS RECOMENDADAS PARA A TODOS PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE.....	63
QUADRO 23 – PRÁTICAS ESPECÍFICAS DE GESTÃO DO CONHECIMENTO EM CADA PROCESSO DE DESENVOLVIMENTO DE SOFTWARE.....	66

SUMÁRIO

AGRADECIMENTOS.....	ii
RESUMO	iv
ABSTRACT.....	v
LISTA DE FIGURAS.....	vi
LISTA DE QUADROS.....	vii
1 INTRODUÇÃO	10
1.1 Justificativa	11
1.2 Definição e delimitação do problema	11
1.3 Objetivos	12
1.3.1 Objetivo geral.....	12
1.3.2 Objetivos específicos	12
1.4 Metodologia	12
1.4 Estrutura do Trabalho	13
2 REVISÃO BIBLIOGRÁFICA.....	14
2.1 Desenvolvimento de Software.....	14
2.1.1 Análise e definição de requisitos	17
2.1.2 Projeto de sistemas e de software	18
2.1.3 Implementação e teste de unidade.....	19
2.1.4 Integração e teste de sistema.....	19
2.1.5 Operação e manutenção	20
2.2 Gestão do Conhecimento	20
2.2.1 Conversões do conhecimento	22
2.3 A gestão do conhecimento no processo de desenvolvimento de software	26
2.3.1 Uma alternativa na distinção do conhecimento	29
2.4 Práticas e ferramentas para a gestão do conhecimento	30
3 DESENVOLVIMENTO	34
3.1 Desenvolvimento do modelo e das práticas e ferramentas.....	34
3.2 Gestão do conhecimento na análise de requisitos.....	36
3.2.1 Práticas e Ferramentas de GC na análise de requisitos.....	37
3.3 Gestão do conhecimento no projeto de sistemas	41
3.3.1 Práticas e Ferramentas de GC no projeto de sistemas	42
3.4 Gestão do conhecimento na implementação	45
3.4.1 Práticas e Ferramentas de GC na implementação	47
3.5 Gestão do conhecimento na integração.....	49
3.5.1 Práticas e Ferramentas de GC na integração.....	51
3.6 Gestão do conhecimento na manutenção.....	53

3.6.1 Práticas e Ferramentas de GC na manutenção.....	55
3.7 Análises e modelos das conversões de conhecimento.....	57
3.8 Análises das práticas e ferramentas	60
4 CONSIDERAÇÕES FINAIS	68
4.1 Contribuição.....	68
4.2 Dificuldades e Limitações	68
4.3 Trabalhos Futuros.....	69
REFERÊNCIAS.....	70

1 INTRODUÇÃO

Na área de desenvolvimento de software um fator importante para que tudo seja realizado com qualidade, é o conhecimento de como realizar cada fase do processo. Para Aurum *et al.* (2007), o processo de desenvolvimento de software é totalmente baseado em conhecimento, fazendo-se assim necessária uma análise mais assertiva da gestão desse conhecimento nas empresas de software.

Considerando o desenvolvimento de software como um serviço, as informações geradas e o conhecimento adquirido na empresa se tornam seus principais ativos. Segundo Nonaka (1991) o único bem certo competitivo para as empresas é o conhecimento. Assim, é definido que as informações geradas em um processo dentro da empresa possuem um alto valor.

A gestão do conhecimento é um paradigma em qualquer que seja o setor da empresa e a área de aplicação, pois os processos de gestão do conhecimento são novos e demandam uma mudança de cultura de toda a empresa, desde a diretoria até o operacional. Expandindo esses paradigmas para as empresas de software pode-se afirmar que temos um agravante, pois toda a criação do produto de software tem total dependência do conhecimento da equipe.

Aurum *et al.* (2007) afirma que quando a gestão do conhecimento é aplicada no processo de desenvolvimento de software os desenvolvedores serão motivados à descreverem suas lições e aprendizados, fazendo com que eles possam trabalhar mais eficientemente e reduzirem retrabalhos.

Logo, a gestão do conhecimento dentro dos processos de software se mostra como uma ferramenta que pode trazer benefícios em questão de metodologias e procedimentos, fazendo com que o entendimento dos processos e o aprendizado sejam mais eficientes.

1.1 Justificativa

O processo de desenvolvimento de software é extremamente sensível ao conhecimento inerente à organização. Tal processo exige que se conheçam as regras de negócios do cliente, as restrições legais e comerciais, as limitações da equipe, as técnicas necessárias para cada fase como programar, testar, documentar, entre outros.

Herbsleb e Moitra (2001) afirmam que as últimas décadas foram determinantes para a observação da necessidade de sistemas e softwares nas empresas de hoje. Utilizar uma ferramenta de software se tornou vital para qualquer tipo de empresa e em qualquer ramo de atividade.

Isso aponta que quanto maior for o conhecimento da empresa, melhor serão seus produtos, menores serão os custos e melhor será gerenciada. A forma como o conhecimento é gerenciado dentro da empresa é extremamente importante para as empresas.

Esse trabalho apresenta uma análise de quais são as necessidades de conversões de conhecimento nos processos de desenvolvimento de software, assim como quais ferramentas e práticas que facilitam tais conversões de conhecimento.

1.2 Definição e delimitação do problema

O trabalho tem como objetivo definir a gestão do conhecimento como uma estratégia eficiente no processo de desenvolvimento de software para apoiar no tempo do processo e na qualidade do software.

O trabalho não busca resolver um problema específico, mas sim conceituar uma possibilidade e definir procedimentos para depois verificar quais são os problemas no processo de desenvolvimento de software que serão beneficiados.

Não existem muitos modelos estruturais que conceituam a gestão do conhecimento no processo de desenvolvimento de software. A maioria dos trabalhos descrevem aplicações práticas ou muito conceituais.

O objetivo deste trabalho é apresentar uma abordagem para a construção de um modelo inicialmente teórico, para depois ser aplicado na prática.

1.3 Objetivos

1.3.1 Objetivo geral

Analisar a gestão do conhecimento no processo de desenvolvimento de software.

1.3.2 Objetivos específicos

Os objetivos específicos são:

- Revisar a literatura sobre o processo de desenvolvimento de software e gestão do conhecimento;
- Identificar as práticas da gestão do conhecimento;
- Integrar as práticas no processo de desenvolvimento de software.

1.4 Metodologia

Este trabalho é de caráter descritivo, e possui como base pesquisas bibliográficas para definições teóricas de procedimentos e processos.

O desenvolvimento do trabalho envolveu os seguintes passos:

- Análise de pesquisas e projetos no ramo com objetivos similares para montar

- uma síntese estruturada;
- Análise de textos e livros que tratem do assunto para desenvolvimento dos conceitos e das práticas;
 - Descrição dos métodos pertinentes e possivelmente aplicados aos processos de desenvolvimento de software;
 - Construção de modelos de conversões de conhecimento para cada fase do processo;
 - Definição de práticas e ferramentas para cada conversão de conhecimento em cada processo.

1.4 Estrutura do Trabalho

No Capítulo 1, é contextualizada a problemática e quais as delimitações do trabalho, mostrando quais são as principais necessidades da gestão do conhecimento no processo de desenvolvimento de software.

No Capítulo 2, é apresentada a revisão bibliográfica, iniciando com as definições do processo de desenvolvimento de software, mostrando cada processo, quais suas funções e importâncias no contexto geral. Além disso, tem o objetivo de apresentar nas seções seguintes quais as conversões existentes para a gestão do conhecimento, como são tratadas essas conversões nos processos de software e quais as necessidades. Ao final do capítulo, são apresentadas as principais ferramentas e práticas da gestão do conhecimento.

O desenvolvimento é realizado no Capítulo 3, onde são contextualizadas as conversões com os processos. Ao final do capítulo, é apresentado o modelo de conversões de conhecimento para o processo como um todo e para cada processo especificadamente, além de mostrar um quadro com todas as ferramentas e práticas citadas para o processo.

No Capítulo 4, são concluídos os pontos do trabalho, apontando as contribuições, as limitações e os problemas e os trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 Desenvolvimento de Software

O IEEE *Computer Society* (SWEBOK, 2004) define a engenharia de software como uma abordagem sistemática, disciplinada e quantificada para o desenvolvimento, operação e manutenção de softwares e suas aplicações.

Sommerville (2003) define como uma área da engenharia que se ocupa com todos os aspectos da produção de um software, desde a concepção até a manutenção dos mesmos. A engenharia de software também se preocupa com o controle e com os processos de softwares. Segundo Sommerville (2003), existem quatro processos fundamentais de software: especificação, desenvolvimento, validação e evolução do software.

Bauer (apud Pressman, 2004) define engenharia de software como "a criação e a utilização de sólidos princípios a fim de obter softwares econômicos para trabalharem eficientemente em máquinas reais".

Baetjer (apud Pressman, 2004) afirma que o processo de desenvolvimento de software é um aprendizado social, considerando que todo conhecimento da empresa está na forma tácita, disperso e incompleto na sua totalidade, e um processo fornece a iteração desse conhecimento, permitindo um diálogo constante de uma fase para outra.

Os processos de softwares são geralmente complexos, considerados intelectuais e dependentes do julgamento humano. Assim, delimitar uma estruturação não é tão simples, o que resulta em vários modelos diferentes de apresentar e gerenciar os processos de desenvolvimento de software.

Um desses modelos de desenvolvimento de software é o modelo em 'cascata'. Esse

modelo foi originado de outros processos de engenharia (apud Royce, Sommerville, 2003). Assim, a estruturação alinhada, procedimental e em linha traz muita semelhança com processos e procedimentos de alguns projetos, desenhos industriais, plantas empresariais, dentre outros.

Para Pressman (2004) o modelo em cascata sugere uma abordagem sistemática e sequencial, iniciando-se na especificação de requisitos e progredindo para a modelagem, construção, implantação e terminando na manutenção progressiva do software criado.

Boehm (1988) afirma que o modelo em cascata se tornou a base da maioria dos modelos de desenvolvimento da atualidade. Isso porque ele foi um dos primeiros a serem definidos e apresentados na literatura.

Os principais estágios do modelo são retratados pelas atividades fundamentais sendo dispostos em fases, onde uma fase só pode ser iniciada quando a fase anterior se termina. As fases desse modelo são definidas por Sommerville (2003) como :

1. Análise e definição dos requisitos: nessa fase, as funções e restrições do sistema são definidas por meio de consultas ao usuário e envolvidos, possuindo ao final as especificações detalhadas do sistema.
2. Projeto de sistemas e de software: logo após possuir todas as especificações e restrições, passamos para a fase na qual serão abstraídas essas definições em requisitos técnicos e lógicos, permitindo um entendimento para que isso seja implementado pelas lógicas de programação. Nessa fase, são elaborados diagramas e fluxogramas.
3. Implementação e teste de unidade: nesse estágio as abstrações feitas nos processos anteriores são implementadas em códigos (definidos no primeiro estágio) e, esses, assim que implementados, são testados unitariamente para a verificação do atendimento dos requisitos.
4. Integração e teste de sistema: após testar os sistemas unitariamente eles são integrados para um teste final e generalizado, a fim de verificar se o sistema como um todo está atendendo a todos os requisitos do cliente. O sistema é entregue ao cliente após essa fase.

5. Operação e manutenção: geralmente é uma das maiores fases do ciclo de vida do software. O sistema é colocado em operação. Assim, a manutenção tem como objetivo resolver possíveis problemas que apareçam na utilização do sistema, além da descoberta de novos requisitos para incrementar o sistema.

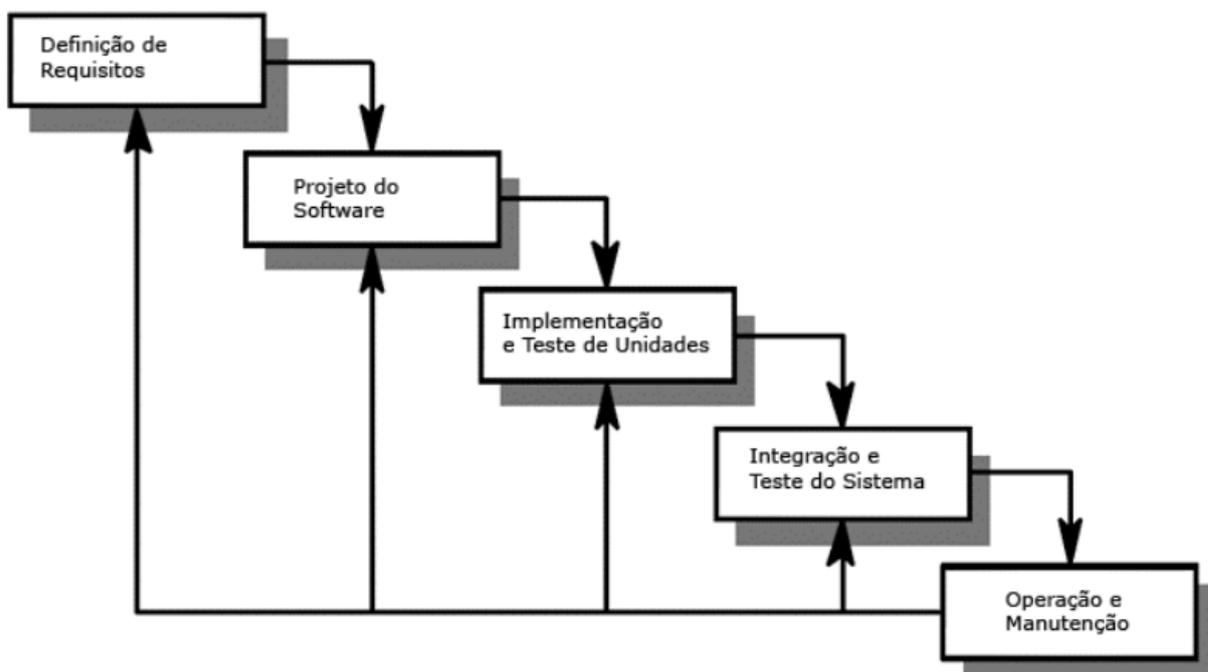


Figura 1 - Representação do modelo em cascata
Fonte: (SOMMERVILLE, 2003, p.).

A Figura 1 mostra a representação do modelo em cascata, na qual as setas acima do processo são as transições de fases e entregas de documentos pertinentes para o início da próxima fase. As setas abaixo dos processos são a documentação e o *feedback* dos processos sucessores para os anteriores.

Sommerville (2003) também descreve que durante esses processos eles devem trocar informações, fazendo com que falhas sejam identificadas como, por exemplo: durante a codificação são encontrados erros no projeto de sistema e de software. Com isso, muitas fases acabam por ter de se repetirem para o ajuste das fases anteriores. Esses processos não são lineares, funcionando por iterações, e com os ajustes de cada fase anterior as iterações acabam ficando mais constantes e repetitivas, trazendo como consequência a suspensão de várias fases do

desenvolvimento para 'adiantar' o processo e dar prosseguimento nas atividades. As suspensões constantes dos processos fazem até com que o sistema seja entregue no prazo. Essa ação traz como consequência erros e omissões nos requisitos originais do software.

A manutenção do software pode fazer com que o sistema seja repensado e novos requisitos venham a aparecer. Assim, todo o processo em cascata pode iniciar novamente no mesmo software, mas com outras finalidades e requisitos. Como falha, Sommerville (2003) descreve que o processo de desenvolvimento em cascata é muito inflexível na sua divisão de tarefas. Isso faz com que o processo não acompanhe as mudanças constantes dos clientes. Este processo geralmente é utilizado quando os requisitos são claros e suficientes.

2.1.1 Análise e definição de requisitos

Conforme o IEEE *Computer Society* (2004) um requisito é definido como uma propriedade que deve ser apresentada para resolver algum problema real.

Sommerville (2003) define quatro fases principais no processo de análise de requisitos: estudo de viabilidade, levantamento e análise de requisitos, especificação de requisitos e validação de requisitos.

O IEEE *Computer Society* (2004) também define que a área de estudo da engenharia de requisitos se divide em quatro processos: elicitação, análise, especificação e validação dos requisitos de software.

Bell e Thayer (apud Lamsweerde, 2000) observaram que há muitos requisitos inadequados, inconsistentes, duplicados ou incompletos nos projetos e possuem um impacto crítico na qualidade do software final.

Assim, entende-se que a primeira fase do processo é considerada uma das mais importantes pelos autores, para alguns até a mais importante, fazendo com que seja

necessária uma atenção em sua excelência, para não acarretar em prejuízos para todo o projeto.

2.1.2 Projeto de sistemas e de software

O projeto de sistemas e de softwares, ou design de software, é composto por descrições estruturais do software para depois de montada ser implementada. É uma fase intermediária do desenvolvimento de sistema, já que vários arranjos lógicos e definições arquiteturas precisam ser determinadas antes de serem desenvolvidas.

De acordo com a IEEE *Computer Society* (2004), o processo de design de software pode ser definido como a arquitetura, os componentes, as interfaces, e demais características de um sistema.

Guindon (1990) aponta que o processo de design de software tem como funcionalidade transformar as informações, geralmente, ambíguas e incompletas, em um design de sistema de alto nível para fácil compreensão dos desenvolvedores.

Segundo Fairley (apud Guindon, 1990), o design de software descreve todas as funções, modelos, fluxos de informações, estruturação de dados e interfaces do sistema.

Sommerville (2003) afirma que o processo de software pode envolver o desenvolvimento de vários modelos em diferentes níveis de abstração e que o projeto de software possui atividades específicas: projeto de arquitetura, especificação abstrata, projeto de interface, projeto de componentes, projeto de estrutura de dados e projeto de algoritmos.

2.1.3 Implementação e teste de unidade

O termo de construção ou implementação de Software, segundo o IEEE *Computer Society* (2004), se refere ao trabalho de criação detalhada do software, através de codificação, verificação, testes de unidade, testes de integração e depuração.

Segundo Sommerville (2003), a programação de sistema é algo pessoal, não existe um processo geral que seja normalmente seguido. Em muitos casos, alguns programadores começarão desenvolvendo os componentes, em outros começam desenvolvendo a interface e outros ainda fazem as especificações só no final do desenvolvimento.

Os desenvolvedores podem testar seus próprios códigos, já colaborando para a próxima fase do processo, mas o objetivo principal do desenvolvedor é verificar se seus processos e procedimentos não possuem defeitos. Esses testes são comumente chamados de depuração.

2.1.4 Integração e teste de sistema

Segundo o IEEE *Computer Society* (2004), a fase de teste é a atividade na qual é avaliada a qualidade do sistema, podendo aperfeiçoá-lo conforme feita a identificação de erros e problemas.

Em alguns casos de desenvolvimento, o teste é tratado apenas como a verificação do código e das funções que ele executa.

Para Sommerville (2003), os estágios do processo de testes são: teste de unidade, teste de módulo, teste de subsistema, teste de sistema e teste de aceitação.

Para Bertolino (2007), o teste de software envolve uma gama de atividades diferenciadas que podem ser validações de linhas de códigos, verificações dos atendimentos de requisitos, ou então verificação do tempo de execução e iteração

com a rede externa,. Assim, sendo necessário uma análise mais procedural dos testes, para haver uma documentação adequada e um controle mais próximo.

2.1.5 Operação e manutenção

Para Sommerville (2003), as alterações de software, por mais que já sejam onerosas, são muito mais baratas do que as alterações de hardware. Isso gera uma motivação nas empresas a fazerem alterações constantes nos softwares.

Existem vários paradigmas sobre manutenção de software, fazendo com que esse processo seja encarado muitas vezes discriminadamente. O processo de manutenção tem importância para a evolução do software, viabilizando a interação do sistema com o usuário e gerando a possibilidade de um *feedback* do que acontece no sistema e se suas expectativas estão sendo atendidas.

Anquetil et al (2007) afirmam que a manutenção dos softwares não é um problema, uma vez que não pode ser evitada já que faz parte da evolução do sistema, e é o que permite a adequação aos requisitos não atendidos ou ao crescimento frente à novos requisitos.

Sommerville (2003) deixa claro que esse limite está desaparecendo, pois os sistemas não estão mais sendo considerados definitivamente como novos. Os softwares atuais estão sempre em evolução. Isso demonstra que quando o processo é colocado em operação, os *feedbacks* já são colocados como novos requisitos, fazendo com que todo o processo de desenvolvimento se inicie novamente para as incrementações do software.

2.2 Gestão do Conhecimento

O conhecimento, conforme Davenport e Prusak (1998), é uma mistura de

experiências, valores, informações, *insights* e contextos que proporciona uma base para avaliação e incorporação de novas experiências e informações.

Para Rao (2005), o século 21 está marcado por um crescimento enorme na economia, com uma política complicada e inúmeras inovações acontecendo em todo o mundo. Nesse contexto, a gestão do conhecimento está sendo encarada como a disciplina crucial para apoio na gestão de riscos, no crescimento da produtividade, na redução de custos, na retenção do conhecimento das empresas e no aumento da inovação.

Laudon e Laudon (2004) afirmam que o conhecimento tomou tamanha relevância para as empresas que a competência na sua gestão pode conduzi-las ao sucesso ou ao fracasso.

Hoog, Spek e Wiig (1997) listam algumas características do conhecimento que mostram a necessidade de gerenciá-lo:

- o conhecimento é intangível e difícil de mensurar;
- o conhecimento é volátil, podendo "desaparecer" do nada na empresa;
- o conhecimento, na maioria das vezes, está inserido em indivíduos com vontades próprias;
- o conhecimento não é processado nos sistemas, ele cresce com o uso constante;
- o conhecimento tem grande impacto nos resultados das organizações;
- o conhecimento não pode ser simplesmente comprado;
- o conhecimento pode ser utilizado por diferentes processos ao mesmo tempo.

Davenport e Prusak (1998) diferenciam quatro elementos para dar apoio na definição de conhecimento: dado, informação, conhecimento e ação. Os dados são fatos distintos, sem representatividade, apresentando apenas um tipo de informação. A informação é um dado inserido em um contexto. O conhecimento é um conjunto definido com informações, contextos e valores pessoais dos indivíduos. As ações são atitudes baseadas nos conhecimentos.

Nonaka e Takeuchi (2008) dividem e definem o conhecimento de duas formas, o conhecimento tácito e o conhecimento explícito.

O conhecimento explícito é aquele que pode ser facilmente expresso em palavras, números ou sons, podendo ser compartilhado em forma de dados, fórmulas, gráficos, entre outros. Esse conhecimento pode ser facilmente formalizado e disseminado.

O conhecimento tácito não pode ser facilmente descrito em palavras, pois é baseado em crenças, experiências, conceitos e aprendizados dos indivíduos. Esse conhecimento não é fácil de ser formalizado por ser pessoal, pois está enraizado nas ações e na experiência do indivíduo.

Segundo Nonaka e Takeuchi (2008), o conhecimento tácito pode ser dividido em duas dimensões. A primeira dimensão é a "técnica", na qual engloba as habilidades formais, o *know-how* dos indivíduos. A segunda dimensão é a "cognitiva", que consiste em crenças, percepções, ideais, valores, modelos mentais que os indivíduos consideram naturais.

Para uma análise das duas representações de conhecimento, pode-se aferir que não existe conhecimento tácito ou explícito, mas sim uma junção dos dois tipos. Aparentemente eles são opostos, mas assim como Nonaka e Takeuchi (2008) apresentam, "o conhecimento é paradoxal, pois é formado do que aparenta ser dois opostos".

2.2.1 Conversões do conhecimento

Nonaka e Takeuchi (2008) afirmam que uma organização cria e utiliza conhecimento convertendo o conhecimento tácito em explícito e vice-versa. A partir disso, são identificados quatro modelos de conversão de conhecimento:

- **Socialização:** de tácito para tácito. Nessa forma o conhecimento tácito é compartilhado e criado através de experiência direta.

- Externalização: de tácito para explícito. O conhecimento tácito é articulado através do diálogo e da reflexão para a construção de modelos e processos.
- Combinação: de explícito para explícito. Nesse modelo o conhecimento explícito é sistematizado e agrupado com outros conhecimentos explícitos.
- Internalização: de explícito para tácito. O conhecimento tácito é aprendido na prática através da disponibilização de conhecimentos explícitos.

Esse modelo é conhecido na literatura como modelo SECI, espiral SECI ou processo SECI. A Figura 2 mostra o modelo SECI. Existe uma espiral que faz com que o crescimento esteja em ascensão, começando pela socialização. O nível de abrangência também é algo a ser pontuado, pois começando na socialização o foco maior é no indivíduo, e assim vai expandindo, sendo que na nova socialização o foco volta para o indivíduo.

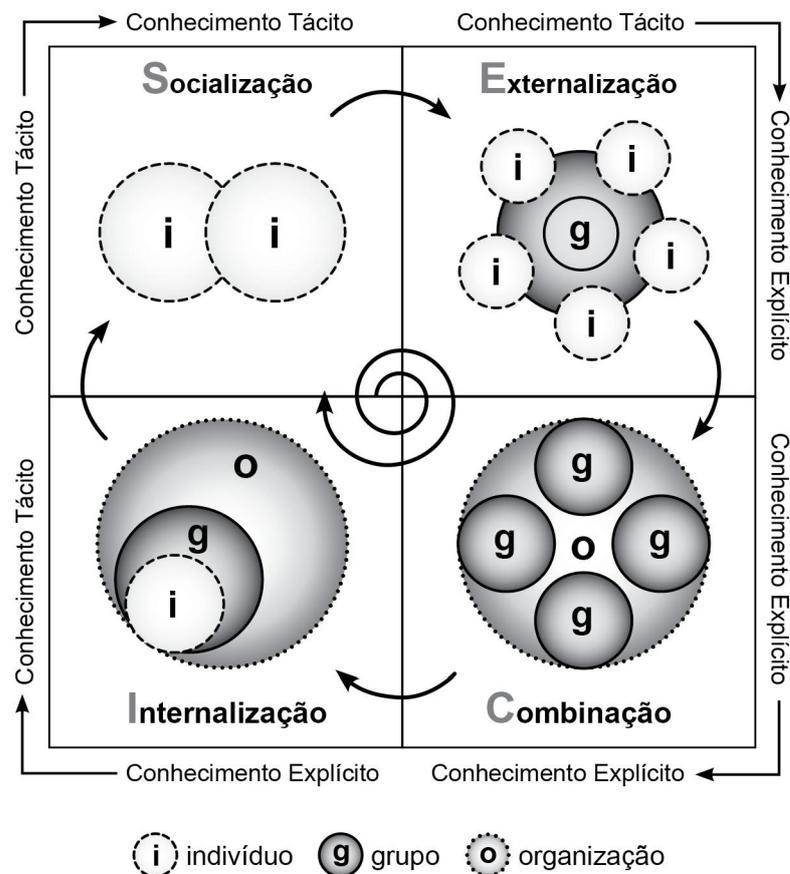


Figura 2 - Modelo SECI
Fonte: (KONNO; NONAKA, 1998, p.).

O conhecimento é amplificado conforme vai caminhando por cada processo, pois a cada processo vai se desenvolvendo, assim como cada modo envolve uma combinação diferente das entidades. A socialização é combinada de indivíduo para indivíduo. A externalização de indivíduo para grupo. Já a combinação é do grupo para a organização. E a internalização é da organização para o indivíduo.

A socialização, conforme Nonaka e Takeuchi (2008) descrevem, é um processo de compartilhamento de experiências, fazendo com que o conhecimento tácito seja criado.

No mundo corporativo, o treinamento do trabalho, acompanhado por um *feedback* constante, usa esse mesmo princípio de socialização. A base do desenvolvimento do conhecimento tácito é trazida pela experiência e o compartilhamento das experiências de trabalho podem fazer o processo de crescimento muito mais eficaz.

Silva (2004) afirma que a socialização acontece quando: existe o diálogo "face-a-face"; os processos de *brainstorming*, *insights* são disseminados por grupos heterogêneos; valoriza-se o trabalho "mestre-aprendiz"; e há compartilhamentos de experiências e modelos mentais via trabalho em equipe.

Para Andrade (2005), a socialização consiste no compartilhamento de experiências por meio da observação, imitação e prática.

No desenvolvimento de produtos para clientes, a socialização está presente também, pois essas interações são processos de compartilhamento do conhecimento tácito e de criação de ideias para a melhoria.

Nonaka e Takeuchi (2008) afirmam que a externalização é o processo chave para a criação do conhecimento. Esse processo é a articulação do conhecimento tácito para o conhecimento explícito, fazendo com que ideias, conceitos, crenças e modelos se tornem descrições lógicas, concisas e coesas. Ele geralmente é desencadeado pelos diálogos e reflexão coletiva, sendo totalmente dependente do grupo para a sua existência.

Andrade (2005) descreve que a externalização acontece por meios de metáforas, analogias, conceitos, hipóteses e modelos. Para Silva (2004), a externalização também acontece por meios de representações simbólicas através de modelos, conceitos, hipóteses, planilhas, textos, imagens, figuras, regras, *scripts*, UMLs, gravações de relatos orais e/ou imagens de ações.

A combinação é um processo de agrupamento de informações por meio de separação, adição, combinação e classificação do conhecimento existente. Geralmente são sistematizados por meio de reuniões, conferências, telefonemas e outros meios computadorizados.

Para Andrade (2005), a combinação "envolve a reconfiguração das informações existentes através da classificação, do acréscimo, da combinação e da categorização do conhecimento explícito".

Silva (2004) descreve a combinação como um agrupamento dos registros de conhecimento que normalmente acontece por meio de junção e processamento de diferentes conhecimentos explícitos.

A internalização é o processo de conversão de conhecimento explícito em tácito. Para Nonaka e Takeuchi (2008), a internalização está intimamente ligada ao "aprender fazendo". Nesse processo, todo o conhecimento que está estruturado em diagramas, textos e informações detalhadas é colocado em ação pelos indivíduos, tanto para aprendizado do que está escrito como para validar o conhecimento explícito.

Andrade (2005) afirma que a internalização ocorre sob a forma de modelos mentais e *know-how* técnico compartilhado. Conforme Silva (2004) apresenta, a internalização acontece por meio de leituras, visualizações e estudos de documentos de vários formatos, práticas individuais, reinterpretações, experimentações, vivências e práticas individuais.

2.3 A gestão do conhecimento no processo de desenvolvimento de software

Schneider (1998) destaca que o conhecimento auxilia as empresas de software a agirem e reagirem rápido. Nesse ambiente de constante desenvolvimento e crescimento, essa competência é um diferencial no mercado, assim como futuramente poderá ser encarada como algo necessário para garantir a sobrevivência da empresa.

Basili e Oivo (1992) afirmam que técnicas baseadas em conhecimento são promissoras nos aspectos de engenharia de software e que por intermédio disso, alguns projetos conseguiram objetivos ambiciosos com práticas baseadas em conhecimento.

Para Abdel-Hamid (apud Chen et al, 2002), os projetos de desenvolvimento de sistemas possuem muitos desafios, como requisitos mal definidos, hardware e softwares que mudam constantemente e alta rotatividade.

Ayas (apud Disterer, 2002) afirma que gerenciar os projetos através de ferramentas de projetos é necessário, mas não é mais o suficiente, é preciso buscar algo mais. Disterer (2002) aponta que o sucesso do projeto depende fortemente da combinação correta dos conhecimentos e experiências da equipe de desenvolvimento.

Biggerstaff (apud Anquetil et al, 2007) apresenta a necessidade do tratamento correto do conhecimento sobre o domínio da aplicação, podendo isso ser determinante para o sucesso do software. Para isso, o conhecimento sobre o domínio da aplicação deve estar disponível desde o começo do projeto, na elicitação de requisitos.

Para a elicitação de requisitos, em alguns casos os desenvolvedores podem apresentar dúvidas sobre informações específicas dos clientes, tais como conhecimentos técnicos sobre químicas para algum cliente do setor ou conhecimentos da área de medicina para a construção de um sistema para hospitais.

Basili e Oivo (1992) afirmam que técnicas de modelagem baseadas em conhecimento são muito importantes para o gerenciamento do projeto, tanto para a compreensão das modelagens quanto para a automação de métodos repetitivos.

Guindon (1990) aponta que o design de software envolve vários domínios de conhecimento como: área de domínio, arquiteturas de software, ciências da computação, métodos de design, UML, dentre outros. Isso apresenta uma oportunidade grande para a absorção e desenvolvimento na equipe em todos esses domínios.

Na análise, padronizar os documentos e estruturar padrões para a criação de diagramas facilitam a criação dos documentos quando forem analisados os requisitos, assim como podem facilitar o entendimento dos documentos no desenvolvimento do sistema, além de possibilitar a automação de algumas modelagens.

Dois dos fatores na codificação do software colocada como fundamental são a minimização da complexidade e a antecipação à mudança (IEEE Computer Society, 2004). Um dos fatores que beneficiam na minimização da complexidade é o conhecimento aprofundado sobre o projeto e sobre formas de desenvolvimento. Já para antecipar à mudança, Nonaka e Takeuchi (2008) apresentam que as empresas mais adaptáveis e prontas para as mudanças externas são aquelas que conseguem gerenciar efetivamente seu conhecimento.

Dentro do desenvolvimento, os desenvolvedores precisam primeiramente de padrões para seus códigos. Mas existem vários casos em que o compartilhamento das experiências dos programadores mais antigos fará com que o desenvolvimento dos programadores mais novos sejam mais eficientes. Com casos de soluções de problemas registrados, o aprendizado será muito mais eficiente e os erros podem ser menos constantes.

Vegas (apud Bertolino, 2007) aponta que em pesquisas, a escolha dos métodos de testes era feita com base nos conhecimentos anteriores, mas não era documentada

ou padronizada, fazendo com que nem todos da equipe soubessem qual seria o teste correto para o tipo de sistema. Com uma documentação ou padronização, o conhecimento poderia ser disponibilizado mais facilmente.

O conhecimento reduz esforços no processo de testes. O registro e armazenamento de erros fazem com que a equipe tenha o *feedback* dos tipos de falhas que os sistemas geralmente possuem, fazendo com que o número de falhas diminua.

Para Anquetil et al (2007), muitas das dificuldades associadas à manutenção do software, como códigos confusos, desconhecimento do domínio, desconhecimento de ferramentas, entre outras, são originadas pela ausência de uma efetiva gestão do conhecimento.

Jørgensen e Sjøberg (apud Anquetil et al, 2007) mostraram que a manutenção constante nos códigos que apresentam problemas faz com que tais problemas sejam reduzidos nos próximos projetos, mostrando que, conforme o conhecimento das aplicações vai aumentando, os riscos de erros serão minimizados.

Aurum e Ward (2004) concluem em seu trabalho alguns pontos importantes a serem relacionados, como: o fato de gerenciar o conhecimento no processo de desenvolvimento de software garante a competitividade no mercado, os processos de gestão do conhecimento são muito imaturos para uma definição estruturada, os desenvolvedores tem dificuldade em explicar o que eles fazem, como eles fazem e porque eles fazem e a liderança e as ferramentas são os principais facilitadores para os processos.

Com isso, Aurum e Ward (2004) definem que, para que essa gestão do conhecimento seja eficiente, é necessário um grande apoio da liderança para fazer os processos funcionarem. Além disso, com a dificuldade do trabalho nesses processos, se faz necessário o apoio de ferramentas prontas, o que não impede a criação de uma ferramenta própria, já que a empresa desenvolve sistemas. Entretanto, a criação de uma nova ferramenta custará tempo para a equipe, e isso se constitui, em quase todas as empresas, como uma das maiores limitações.

Avram et al (2009) chega a conclusão de que a cultura dentro da empresa é um fator primordial para o desenvolvimento da gestão do conhecimento nas empresas de software. Mas não implica que a cultura é o único fator a ser levado em consideração.

Feher e Gabor (2006) concluem que empresas de engenharia de software com culturas orientadas para o conhecimento possuem uma vantagem estratégica para a área de soluções organizacionais.

Trabalhar a cultura em uma organização é um processo que precisa de uma grande atenção. Não se pode fazer uma padronização para as definições de cultura, já que mudá-la dentro de uma empresa pode trazer riscos grandes. Cada empresa tem uma cultura diferente, e o trabalho para cada empresa tem que ser específico.

2.3.1 Uma alternativa na distinção do conhecimento

Para Herbsleb e Moitra (2001), no processo de desenvolvimento de software, as informações geradas, os desenvolvedores, os gerentes e os clientes estão cada vez mais diversificados, fazendo com que a compreensão de tais informações possam demandar muito tempo. Assim sendo, é necessária uma gestão efetiva das informações e do conhecimento para que a comunicação no projeto seja eficiente.

Maurer e Holz (1999), para uma distinção melhor das informações e dos conhecimentos nos projetos, fazem uma divisão do conhecimento no desenvolvimento de software como: conhecimento genérico ou conhecimento reusável; conhecimento de projetos específicos; e dados de projetos.

Segundo esses autores, os conhecimentos genéricos são os conhecimentos baseados em tarefas repetitivas, objetivos de tarefas comuns, modelos de processos, procedimentos padrões, entre outros. Esses conhecimentos geralmente são mapeados em algum sistema que disponibiliza as informações necessárias para as execuções das tarefas.

Os conhecimentos de projetos são conhecimentos e informações específicos inerentes aos projetos em questão, como custos, pessoas envolvidas, tarefas específicas do projeto, etc.

Geralmente, os conhecimentos de projetos fazem uma abordagem buscando conhecimento genérico. Exemplificando, em um projeto a fase de teste é tratada como conhecimento em projeto, mas é redirecionada para um conhecimento genérico como "teste unitários", "testes de casos de uso", entre outros.

Os dados de projetos são todos documentos e informações que são criados quando as tarefas são executadas. Geralmente, são os dados específicos que são modelados depois para transformação de conhecimento genérico ou conhecimento de projetos.

Com essa divisão de tipos de conhecimento é mais fácil trabalhar os processos nas empresas, pois os esses são familiares para as equipes. O gerenciamento desses tipos de conhecimento é mais fácil. Ainda assim, se faz necessária a utilização de alguma ferramenta para fazer as divisões dos tipos de conhecimentos e apresentação para a equipe, assim como vinculada ao sistema que a equipe utiliza para gerenciar o projeto.

2.4 Práticas e ferramentas para a gestão do conhecimento

Hoog, Spek e Wiiig (1997) esclarecem deixam claros em seu trabalho que existiam vários textos que tratavam a gestão do conhecimento, todavia alguns apresentavam totalmente o trabalho operacional e outros se apresentavam um conteúdo mais teórico, existindo uma lacuna no que diz respeito às formas transformar a teoria em práticas operacionais.

Existem várias práticas e ferramentas que ajudam ou apoiam o desenvolvimento do conhecimento, assim como a sua criação e disponibilização para as organizações.

Com essas ferramentas, o desenvolvimento do conhecimento se torna algo mais fácil e agradável para trabalhar, contudo o uso dessas ferramentas não reduz as necessidades de esforços da alta administração e lideranças para o desenvolvimento da cultura e das competências também.

Buscolo et al (2009) fazem um levantamento de práticas e ferramentas que são utilizadas para a gestão do conhecimento. Não se pode afirmar que existe uma ferramenta exata ou uma receita pronta para a utilização dessas ferramentas, mas tê-las em mente fará com que haja uma abertura às utilizações.

As práticas mais citadas na literatura e artigos, conforme Buscolo et al (2009), são:

- Trabalhar juntamente à alta diretoria;
- Mapear as habilidades e competências dos funcionários;
- Criar uma conectividade entre as pessoas;
- Fazer um aperfeiçoamento do layout dos escritórios;
- Criar um programa de mentores;
- Incentivar através políticas de RH;
- Incentivar a geração de ideias;
- Incentivar a participação em congressos e seminários;
- Implementar ferramentas adequadas de TI.

Werner e Werner (2000) também citam algumas ferramentas de tecnologia da informação como *Data Warehouse*, *Data Mining*, *Business Intelligence*, *Workflow*, SIG (Sistema de Informação Gerencial), ERP (*Enterprise Resource Planning*), E-Commerce, E-Business e E-Publish, *intranets* e *extranets*, colocando à parte os CMKS (*Customer Management Knowledge System*), ou seja, sistemas específicos para a gestão do conhecimento. Werner e Werner (2000) dividem essas ferramentas de tecnologia em grupos: armazenamento, disponibilização e descobrimento.

No armazenamento, o foco está nas ferramentas de SGBD (sistemas de gerenciamento de banco de dados), *Data Mining*, *Data Warehouses*, etc., sendo o seu objetivo principal o de guardar, organizar e disponibilizar o conhecimento de forma adequada. Na disponibilização, o enfoque está nas *intranets* e demais

sistemas que disponibilizem esse conhecimento em um formato que seja facilmente identificado e processado pelos indivíduos. No descobrimento, encontram-se as ferramentas de *Data Mining* e *Text Mining*, para a descoberta do conhecimento que existe na empresa, mas as pessoas não sabem que a organização possui tal conhecimento.

Costa (2005) classifica as ferramentas de gestão do conhecimento como:

- Ferramentas de socialização: permitem a integração de indivíduos, como e-mails, salas de bate papo, fóruns, etc;
- Ferramentas de captura de conteúdo: auxiliam a transformar conhecimento tácito em explícito, como sistemas dinâmicos de FAQ (questões frequentes perguntadas), sistemas de erros reportados, etc;
- Ferramentas de informação: auxilia na busca, filtragem e uso das informações na empresa, podendo ser útil na busca de palavras-chaves em documentos, ou na busca de pessoas com conhecimentos em determinados assuntos;
- Ferramentas de evolução da informação e conhecimento: permitem a qualificação dos conteúdos na empresa, auxiliando nas reutilizações de conhecimentos ou descartes de conhecimentos e informações;
- Ferramentas de infraestrutura: permitem a existência dos sistemas, como banco de dados, sistemas operacionais, rede de computadores, etc.

As ferramentas de socialização fazem um *link* com os funcionários, permitindo a interação e o desenvolvimento do conhecimento através da troca de experiências. As ferramentas de captação de conteúdo fazem com que, dentro das ferramentas de socialização, seja possível conceituar os conhecimentos tácitos em explícitos. As ferramentas de informação tratam os dados externos e internos da empresa, criando relatórios e/ou listas para serem interpretados. As ferramentas de evolução permitem que o conhecimento não fique obsoleto e esteja sempre em evolução, pois caso algum conhecimento esteja ultrapassado, ele é qualificado assim, podendo ficar visível para quem o reutilize. E as ferramentas de infraestrutura são a base para o desenvolvimento das demais, permitindo que os dados, informações e conhecimento fiquem corretamente armazenados e seguros na empresa.

O recomendado pelos autores é trabalhar com as práticas e ferramentas em conjunto, pois se o foco for só atuar com práticas, a aplicação ficará complexa demais, e sem as ferramentas para apoio, as análises ficarão subjetivas. Da mesma forma, se a atuação for só com as ferramentas, elas ficarão paradas na empresa, sem uso, já que a organização não se preparou para utilizar essas ferramentas e nem encontram motivações em usá-las.

3 DESENVOLVIMENTO

3.1 Desenvolvimento do modelo e das práticas e ferramentas

O desenvolvimento é baseado na construção de uma análise prática que possa ser uma ponte para a teoria das conversões de conhecimento e as práticas possíveis no ambiente de desenvolvimento de sistemas na empresa.

O modelo em cascata, Figura 3, faz uma abordagem das transições de estágios por meio de documentos e informações que dão sinais para a execução do próximo estágio. Além de representarem o *feedback* existente da operação e manutenção para os estágios anteriores.

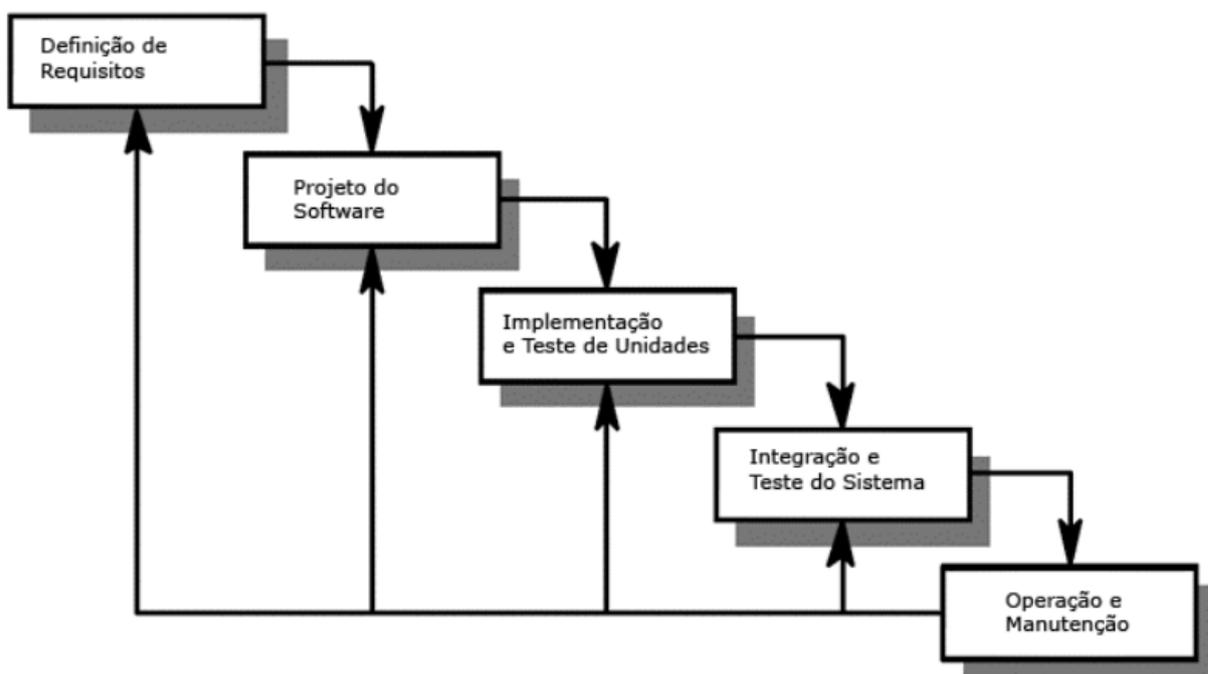


Figura 3 - Representação do modelo em cascata
Fonte: (SOMMERVILLE, 2003, p.).

Toda a abordagem do modelo em cascata é focada nas transições de informação e dados que são tratados e desenvolvidos. Com base na estrutura do modelo cascata, será desenvolvido como as conversões de conhecimento serão tratadas em cada processo.

Serão modelados em cada fase do processo de software, quais as conversões possíveis e quais são os envolvidos nessas conversões, e, para cada conversão, serão apresentadas quais as práticas e ferramentas que possibilitarão tal conversão.

Ao final, será definido um modelo base para análise das conversões de conhecimento em todo o processo em cascata, podendo ser interpretado em todo o modelo como é possível converter os conhecimentos em todo o processo. E também com quais práticas e ferramentas isso será realizável.

Delimitação de Ferramentas por Grupos	
Ferramentas de Gestão do Conhecimento (Costa, 2005)	Ferramentas de Tecnologia da Informação. (Werner e Werner, 2000)
Ferramentas de Socialização	<ul style="list-style-type: none"> • Chats; • Fóruns; • E-mail;
Ferramentas de Captura de Conteúdo	<ul style="list-style-type: none"> • SIG; • ERP; • E-Commerce; • E-Business; • E-Publish; • Intranets e Extranets; • Workflow. • Erros reportados
Ferramentas de Informação	<ul style="list-style-type: none"> • SIG; • ERP; • Intranets e Extranets; • Fóruns.
Ferramentas de Evolução	<ul style="list-style-type: none"> • Data Mining; • Text Mining; • Sistema de suporte à decisão; • Business Intelligence
Ferramentas de Infraestrutura	<ul style="list-style-type: none"> • Sistemas Operacionais; • Data Warehouse; • Servidores;

Quadro 1 – Delimitação de ferramentas de TI com os definidos grupos de ferramentas de Gestão do Conhecimento.

Fonte: Elaborado com base em Costa (2005) e Werner e Werner (2000).

Para uma melhor descrição das ferramentas, serão utilizadas as definições de Costa

(2005), que agrupa as ferramentas em grupos específicos. Para entender melhor esses grupos específicos, será delimitada cada ferramenta apresentada por Werner e Werner (2000) junto aos grupos específicos. Tal delimitação é apresentada na Quadro 1.

3.2 Gestão do conhecimento na análise de requisitos

Na análise de requisitos se tem uma necessidade grande em obtenção de informação e conhecimento, pois qualquer erro nessa fase vai impactar em todos demais processos.

Em quase todos os casos de análises de requisitos, elas são realizadas com reuniões entre o cliente e a empresa. Nessas reuniões são definidas as necessidades do sistema e as possibilidades de desenvolvimento.

A socialização é uma conversão que precisa estar presente no processo, e o afeta diretamente, pois o representante da empresa precisa entender e processar para si todas as informações e conhecimentos que o cliente está passando. O cliente também pode aprender com a socialização, mas o objetivo principal do processo é passar do cliente para o representante da empresa.

A internalização se faz necessária apenas para o aprendizado de ferramentas ou procedimentos que estão de forma explícita dentro da própria empresa, como, por exemplo, modelos de reuniões, perguntas frequentes, entre outros.

Dentro do processo de análise de requisitos, é possível que o conhecimento seja combinado também, podendo ser um grande apoio para a construção dos requisitos. Para isso, é necessário que o cliente inicialmente tenha documentado seus conhecimentos dentro da empresa. Dentro de um servidor, ou banco de dados de conhecimento é possível também combinar os conhecimentos inerentes à análise de requisitos.

À medida que todo o conhecimento vai sendo processado, é extremamente determinante que ele seja externalizado, apontando tal conversão, a externalização, como importante para o processo. Essa externalização deve ser feita para um documento de requisitos, que é o documento final desse processo. É possível externalizar os conhecimentos também para uma base de dados da empresa, para que ele possa ser trabalhado futuramente.

No final desse processo, como documentação e necessidade do modelo em cascata, todos os requisitos e, conhecimentos são externalizados para uma compilação que é tratada aqui como Documento de Requisitos.

A Figura 4 mostra no processo de análise de requisitos quais são as conversões existentes e para onde esse conhecimento é direcionado.

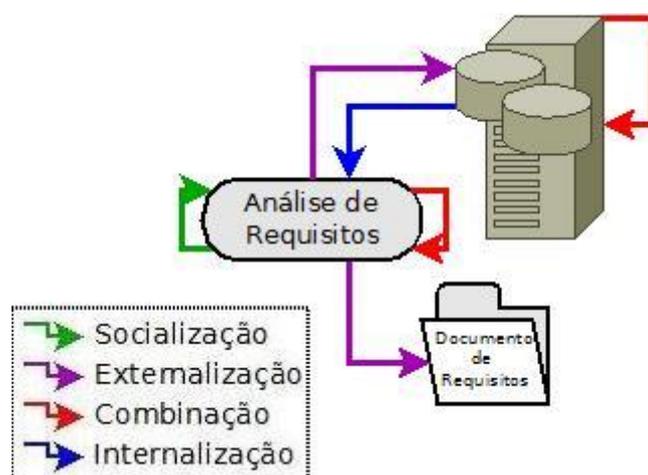


Figura 4 - Conversões do conhecimento na análise de requisitos
Fonte: Autoria própria.

Frente a essas conversões na análise de requisitos, foram estruturadas as práticas e ferramentas para cada conversão de conhecimento.

3.2.1 Práticas e Ferramentas de GC na análise de requisitos

O Quadro 2 apresenta as práticas e ferramentas utilizáveis nos processo de socialização do conhecimento na análise de requisitos.

Socialização na Análise de Requisitos	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Políticas de RH; • Geração de ideias. 	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura.

Quadro 2 – Práticas e ferramentas recomendadas para a socialização na análise de requisitos
Fonte: Autoria própria.

As práticas de socialização podem ter um peso maior processo da análise de requisitos. Manter uma conectividade entre as pessoas faz com que os indivíduos obtenham mais informações e conheçam mais sobre as regras de negócios dos clientes.

Um programa de mentores facilitaria na transição de conhecimento dos mais experientes para os menos experientes, fazendo com que a análise de requisitos possa ser mais bem direcionada, pois serão evitados os erros que foram cometidos no passado.

As políticas de RH vão colaborar no incentivo de geração de ideias e soluções para a necessidade inicial do cliente, bem como incentivar essa geração de ideias dentro da empresa. Por mais que essas ideias não sejam utilizadas, cultivar essa cultura fará com que as boas ideias apareçam mais facilmente.

Já nas ferramentas, o enfoque maior está nas ferramentas de socialização. Tais ferramentas facilitarão a comunicação de todos os participantes, fazendo com que a troca de informações, ideias e conhecimentos seja mais fácil.

As ferramentas de infraestrutura são consideradas já básicas para qualquer empresa que trabalhe com qualquer sistema de informação. Com os sistemas de conhecimento não é diferente. Serão necessários servidores, bancos de dados e um *mainframe* para o fluxo de conhecimento.

No Quadro 3, é identificado como o conhecimento pode ser externalizado na fase de

análise de requisitos.

Externalização na Análise de Requisitos	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH; 	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de evolução; • Ferramentas de infraestrutura.

Quadro 3 – Práticas e ferramentas recomendadas para a externalização na análise de requisitos
Fonte: Autoria própria.

Para a externalização, não foram encontradas muitas práticas direcionadas já que o trabalho é "traduzir" o conhecimento tácito para explícito, e tal procedimento é melhor realizado com as ferramentas adequadas. Na externalização, o conhecimento do processo todo já está conciso, sendo necessária a interpretação e escrita de tais conhecimentos.

Ainda assim, uma prática é considerada importante: o incentivo em políticas de RH, estimulando e apoiando para que o conhecimento seja descrito com qualidade, ou seja, com detalhes e informações pertinentes aos processos. Podendo também tais políticas manter a motivação para o trabalho baseada na gestão do conhecimento.

Sendo as mais requisitadas nesse processo, as ferramentas de captura de conteúdo vão colaborar para serem descritos os padrões e regras, também permitindo a criação de fluxogramas, que são representações de melhor entendimento.

As ferramentas de evolução do conhecimento auxiliam na análise dos requisitos verificando os *trade-offs* existentes, apresentando as redundâncias e classificando essas informações para uma melhor análise na próxima fase. Também são necessárias as ferramentas de infraestrutura nesse processo.

As práticas e ferramentas identificadas na combinação de conhecimentos são definidas no Quadro 4

Combinação na Análise de Requisitos	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Participação em congressos e seminários; • Políticas de RH 	<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura.

Quadro 4 – Práticas e ferramentas recomendadas para a combinação na análise de requisitos
Fonte: Autoria própria.

Dentre as práticas de combinação, tem-se que a participação em congressos e seminários faz com que sejam adquiridos vários conhecimentos externos que podem auxiliar no processo. Esse conhecimento, na maioria das vezes, é apresentado explicitamente, já que o conhecimento tácito vem com a prática.

Incentivar as políticas de RH vai manter o indivíduo focado no seu trabalho de desenvolvimento do conhecimento dentro da empresa.

As ferramentas de infraestrutura são as que mais trabalham com combinações. Já as ferramentas de evolução vão ajudar muito na distinção dos conhecimentos, na seleção correta e no armazenamento. Tal armazenamento que requer ferramentas de infraestrutura.

A internalização na análise de requisitos é estruturada conforme o Quadro 5.

Internalização na Análise de Requisitos	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Trabalho conjunto à diretoria. • Políticas de RH. 	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura.

Quadro 5 – Práticas e ferramentas recomendadas para a internalização na análise de requisitos
Fonte: Autoria própria.

Como descrito anteriormente, a internalização do conhecimento tem como objetivo apresentar para a equipe de desenvolvimento as regras da empresa que estão

externalizadas.

A prática identificada como determinante para essa fase é o trabalho conjunto à diretoria. Nesse caso, a diretoria é interpretada como àqueles que possuem o conhecimento total de seus clientes, possuindo informações vitais e as vezes até confidenciais. Tendo em mãos as informações relevantes aos processos, vai facilitar para que a equipe aprenda mais rapidamente.

As ferramentas de informação podem ajudar na interpretação dos conhecimentos, criando regras e padrões para melhor entendimento, fazendo-se necessário que esse conhecimento explícito esteja já inserido nas ferramentas. Como os demais processos, são necessárias as ferramentas de infraestrutura.

3.3 Gestão do conhecimento no projeto de sistemas

Com o documento de requisitos, o primeiro passo é internalizar todo o conhecimento que está nele. Assim como internalizar os conhecimentos da empresa que são requeridos no processo.

No projeto de sistemas, a socialização é uma conversão que pode fazer com que a qualidade do projeto seja muito maior, pois existem muitos conhecimentos tácitos sobre projetos e gerenciamento de pessoas que podem ser trocados nessa fase.

O trabalho de documentar e estruturar em regras e padrões claros e mais direcionados à programação faz com que a combinação seja requerida nesse processo, pois já se possui os conhecimentos explícitos na análise de requisitos e os conhecimentos explícitos de modelos de projetos, diagramas e representações. Na base de dados da empresa, é necessário combinar os conhecimentos dos projetos já desenvolvidos com os que estão em desenvolvimento.

Ao fim do processo, como característica do modelo em cascata é montado um documento. Nesse caso o projeto do software, com prazos, responsáveis, modelos e

representações do software. Para a elaboração desse documento, é necessário externalizar todo o conhecimento inerente a esse processo, também podendo, quando necessário, externalizar esse conhecimento para a base de dados da empresa.

Na Figura 5 se tem a representação de como é representado esse fluxo de conversões dentro do projeto de sistemas.

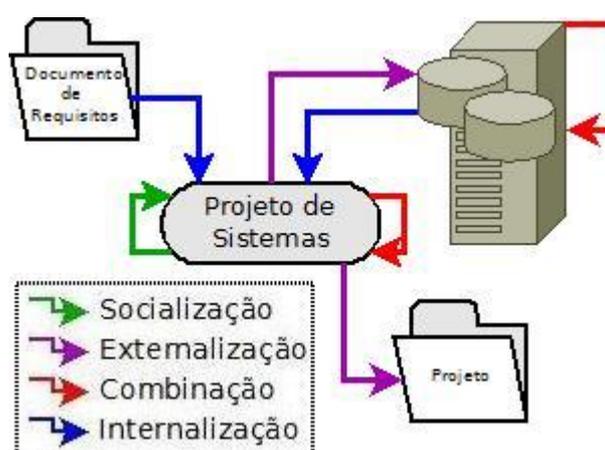


Figura 5 - Conversões do conhecimento no projeto de sistemas.
Fonte: Autoria própria.

3.3.1 Práticas e Ferramentas de GC no projeto de sistemas

Na socialização, representa-se quais são as práticas e ferramentas efetivas para o projeto de sistemas (Quadro 6).

A conectividade entre as pessoas aparentemente faz com que o desenvolvimento do projeto seja desenvolvido mais rapidamente, colaborando com a troca de experiências e de ideias.

As políticas de RH continuarão a dar uma motivação para os indivíduos trabalharem com a gestão do conhecimento nesse processo. A motivação também vai impactar na geração de ideias de cada um.

Socialização no Projeto de Sistemas	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Políticas de RH; • Geração de ideias; • Mapa das habilidades e competências dos funcionários; 	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura.

Quadro 6 – Práticas e ferramentas recomendadas para a socialização no projeto de sistemas
Fonte: Autoria própria.

Criar um mapa de habilidades e conhecimentos dos funcionários fará com que o direcionamento das tarefas seja muito mais eficaz, assim como a garantia de que o foco de cada um estará em suas reais habilidades. Pode ser utilizado também para objetivar o desenvolvimento de novas habilidades, delegando tarefas de pouco conhecimento para alguém que possa se desenvolver nesses tipos de tarefas.

Como percebe-se, as ferramentas de socialização são a base tecnológica para essa conversão de conhecimento, permitindo uma iteratividade e uma comunicação com a equipe. Desta forma, para manter essas ferramentas são necessárias as ferramentas de infraestrutura.

As práticas e ferramentas para externalizar o conhecimento no projeto de sistemas são sugeridas conforme o Quadro 7.

Externalização no Projeto de Sistemas	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH. 	<ul style="list-style-type: none"> • Ferramentas de captação; • Ferramentas de evolução; • Ferramentas de infraestrutura.

Quadro 7 – Práticas e ferramentas recomendadas para a externalização no projeto de sistemas
Fonte: Autoria própria.

Assim como na análise de requisitos, no projeto de sistemas as práticas para a

externalização são mais voltadas às políticas de RH para manter a motivação e garantir o trabalho de todos voltado à gestão do conhecimento.

As ferramentas de captação vão facilitar na elaboração do projeto, assim como na construção dos modelos, estruturas e padrões. As ferramentas de evolução vão auxiliar na análise e desenvolvimento dos modelos, verificando se não existem *trade-offs* e se não existem informações que estão faltando. Para a manutenção das ferramentas anteriores, são necessárias as ferramentas de infraestrutura. Seguindo para a combinação dos conhecimentos, as práticas e ferramentas são descritas no Quadro 8.

Combinação no Projeto de Sistemas	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Participação em congressos e seminários; • Políticas de RH. 	<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura.

Quadro 8 – Práticas e ferramentas recomendadas para a combinação no projeto de sistemas
Fonte: Autoria própria.

As práticas de combinação seguem o mesmo modelo da socialização, através de políticas de RH para a motivação e participação de congressos e seminários que objetivam a obtenção de outros conhecimentos explícitos e aplicação na empresa.

As ferramentas de evolução vão fazer os filtros necessários para as reutilizações e descartes dos conhecimentos existentes e as ferramentas de infraestrutura vão ser a base para as outras ferramentas.

Como uma conversão relevante para o projeto de sistemas, a internalização se faz totalmente necessária para o entendimento dos requisitos e dos projetos anteriores da empresa. Assim, no Quadro 9, são apresentadas as práticas e ferramentas para tal conversão.

Internalização no Projeto de Sistemas	
Práticas	Ferramentas

<ul style="list-style-type: none"> • Políticas de RH. 	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura.

Quadro 9 – Práticas e ferramentas recomendadas para a internalização no projeto de sistemas
Fonte: Autoria própria.

O enfoque das políticas de RH nessa conversão deve ter uma atenção especial para garantir que os indivíduos do processo realmente busquem entender os requisitos da análise anterior.

As ferramentas de informação vão mostrar os conhecimentos no momento certo e da forma correta, auxiliando as pessoas a compreenderem os conhecimentos existentes e permitindo uma identificação melhor de tais conhecimentos.

3.4 Gestão do conhecimento na implementação

Assim que é montado o projeto do sistema, é passado para a próxima fase, a implementação (ou desenvolvimento), sendo obrigação inicial de todos do processo internalizar os conhecimentos do projeto, e, caso necessário, internalizar conhecimentos que possam precisar para desenvolverem o projeto, conforme o que foi descrito.

A socialização é uma conversão que pode ser essencial para a entrega do projeto nos prazos descritos e na qualidade exigida, pois com a troca de experiências, será mais fácil evitar erros já ocorridos e aprender métodos e códigos novos que deixem o trabalho mais eficiente.

A combinação só se faz existente na base de dados da empresa, onde serão tratadas e analisadas as experiências já externalizadas no processo. Dentro desse, não são identificadas possibilidades de combinação, já que, na maioria dos casos, o desenvolvimento é feito em uma ferramenta com uma linguagem pré-definida. Dessa forma, o desenvolvedor limita-se nos conhecimentos da linguagem apenas,

pontuando que as linguagens de programação podem trocar informações entre si, mas os conhecimentos inerentes à cada uma e suas estruturas explícitas são totalmente individuais.

A externalização é representada de três formas. A primeira forma é na construção do código propriamente dito, pois o desenvolvedor transforma seus conhecimentos referentes à linguagem em códigos explícitos. A segunda forma se apresenta na representação de erros e acertos ou na descrição de problemas e soluções, levados para a base de dados da empresa. Já na terceira forma, a externalização aparece na documentação dos códigos, definindo como foram escritos os métodos, para um entendimento das próximas fases.

O documento de Documentação é também um requisito para o modelo em cascata. Já que cada processo é totalmente independente dos demais, é necessário registrar tudo o que foi aprendido e quais as decisões tomadas para que os demais processos saibam como agir e reagir frente a erros.

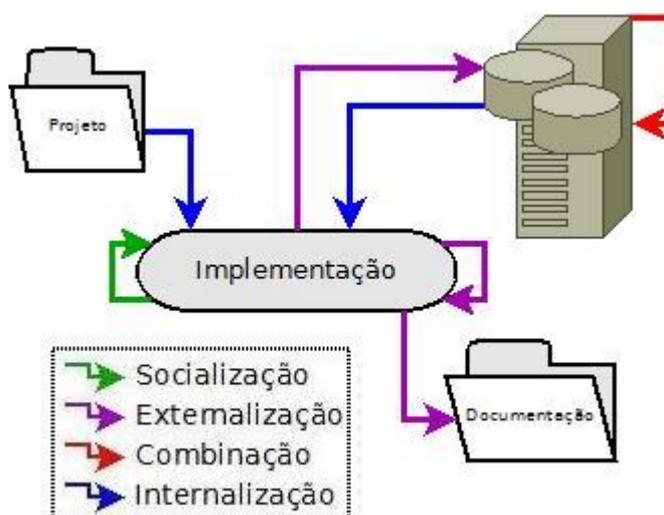


Figura 6 - Conversões do conhecimento na implementação.
Fonte: Autoria própria.

Na Figura 6, verifica-se como representar tais conversões de conhecimento na implementação do software.

3.4.1 Práticas e Ferramentas de GC na implementação

Sendo umas das conversões mais importantes para a implementação, a socialização possui várias práticas envolvidas, assim como as ferramentas já conhecidas. Visualiza-se isso no Quadro 10.

Socialização na Implementação	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Aperfeiçoamento do layout dos escritórios; • Políticas de RH; • Geração de ideias; • Programa de mentores. 	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura.

Quadro 10 – Práticas e ferramentas recomendadas para a socialização na implementação
Fonte: Autoria própria.

A conectividade entre as pessoas faz com que a troca de informações seja mais eficiente e isso pode ser melhorado e otimizado com o aperfeiçoamento do layout dos escritórios, fazendo com que a conectividade entre essas pessoas sejam muito fáceis.

As políticas de RH ainda assim são presentes na implementação para motivar o uso das ferramentas e para a geração de ideias.

A criação de um programa de mentores permite que sejam passadas experiências anteriores de desenvolvimentos para novos funcionários fazendo com que não existam erros recorrentes, podendo os desenvolvedores utilizarem seu tempo com o desenvolvimento do que foi pedido.

As ferramentas de socialização vão permitir que os desenvolvedores troquem informações de códigos, de problemas e soluções, do andamento do projeto, dentre outros. Já as ferramentas de infraestrutura permitem o uso das ferramentas de socialização.

Considerando a externalização na implementação, o Quadro 11 nos mostra as práticas e ferramentas.

Externalização na Implementação	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH; • Participação em congressos e seminários. 	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de infraestrutura.

Quadro 11 – Práticas e ferramentas recomendadas para a externalização na implementação
Fonte: Autoria própria.

As políticas de RH mantém o objetivo de motivação. Já a participação em congressos e seminários pode permitir o aprendizado de novas tecnologias de desenvolvimento, assim, essas tecnologias podem ser aplicadas, e seus códigos externalizados.

As ferramentas de captura também são presentes na externalização na implementação, pois, é necessária alguma ferramenta para documentar os códigos, podendo ser a própria ferramenta de desenvolvimento, assim como são necessárias ferramentas para povoação de erros reportados, problemas e soluções, entre outros.

Combinação na Implementação
Ferramentas
<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura.

Quadro 12 – Ferramentas recomendadas para a combinação na implementação
Fonte: Autoria própria.

Sendo a combinação existente dentro da base de dados da empresa, o Quadro 12 apresenta apenas ferramentas que podem fazer essa combinação.

As ferramentas de evolução vão ajudar na seleção de problemas recorrentes e nas suas classificações, podendo fazer inferências estatísticas de quais problemas

acontecem mais e quais são as possíveis soluções. Como base, as ferramentas de infraestrutura possibilitam tais ferramentas trabalharem.

A internalização, presente no entendimento do projeto e de novas ferramentas, tem suas práticas e ferramentas descritas no Quadro 13.

Internalização na Implementação	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH; 	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura.

Quadro 13 – Práticas e ferramentas recomendadas para a internalização na implementação
Fonte: Autoria própria.

As políticas de RH, novamente aparecem para motivar os usos das ferramentas e entendimento dos processos.

As ferramentas de informação, mantidas pelas ferramentas de infraestrutura, vão permitir o entendimento dos projetos, podendo classificar as tarefas em importância, em prazos ou em outra característica definida. Assim, os desenvolvedores podem analisar quais as suas prioridades, além de facilitar no entendimento do projeto como um todo.

3.5 Gestão do conhecimento na integração

Após desenvolvido todo o sistema, é passado para a fase de integração, que objetiva testar os módulos e códigos desenvolvidos na fase anterior.

Como base inicial, a equipe de testes precisa entender o código, através da internalização, lendo-o. Além disso, caso necessário, também deve internalizar algum tipo de teste que seja requerido conforme for a documentação da fase anterior. Outra necessidade de internalização é a de entender todos os requisitos

dos clientes para verificar o atendimento de tais requisitos ou não.

A socialização ocorre na forma de aprendizado mútuo entre os indivíduos, colaborando mutuamente os conhecimentos que possuem, além de experiências vividas e modelos de testes já testados que funcionam.

Conforme são realizadas as interpretações dos códigos e eles vão sendo testados, são externalizados relatórios para registro, tanto para analisar no projeto quais foram as maiores dificuldades de desenvolvimento, quanto para armazenamento na base de dados da empresa para conhecimento de quais são as necessidades de treinamentos da equipe de desenvolvimento.

Caso existam modelos e formas de testes externalizados na base de testes, é possível combiná-los para tipos de códigos e linguagens de programações, além de combinar os conhecimentos de cada indivíduo para o crescimento do conhecimento da equipe.

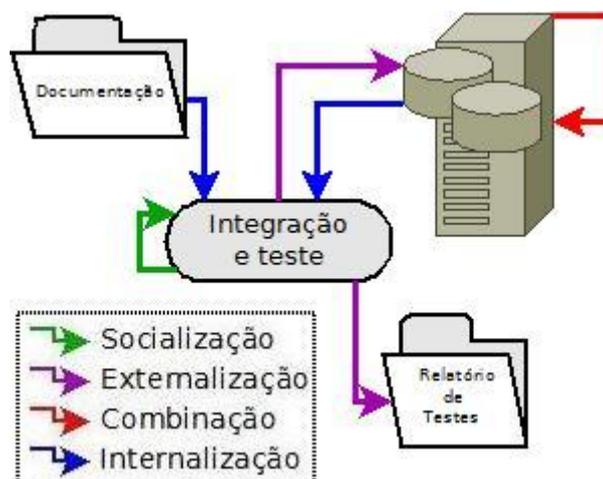


Figura 7 - Conversões do conhecimento na integração.
Fonte: Autoria própria.

O documento de Relatório de Testes contém todos os acontecimentos relatados e as alterações necessárias nos códigos para atendimento dos requisitos, pois tais testes precisam ser reportados para os demais processos.

Na Figura 7, pode-se verificar como ficam representadas essas conversões na fase

de integração.

3.5.1 Práticas e Ferramentas de GC na integração

A fase de testes é um processo de análise dos atendimentos dos requisitos mais focada em encontrar erros e falhas. Pode-se ver a representação da socialização no Quadro 14.

Socialização na Integração	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Aperfeiçoamento do layout dos escritórios; • Políticas de RH; • Geração de ideias; • Programa de mentores. 	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura.

Quadro 14 – Práticas e ferramentas recomendadas para a socialização na integração
Fonte: Autoria própria.

Um dos objetivos principais da socialização nessa fase é facilitar a comunicação para que exista a troca de experiências, e isso é constatado através da conectividade entre as pessoas, além de ter o layout do departamento desenvolvido para essa finalidade. Como nos demais processos, as políticas de RH se fazem presentes para motivação e geração de ideias dos indivíduos.

Já o programa de mentores, tem como objetivo o aprendizado mais rápido dos menos experientes, aprendendo com os mais experientes, já que os mentores terão uma responsabilidade no desenvolvimento desses funcionários menos entendidos.

As ferramentas de socialização vão facilitar a comunicação entre a equipe no desenvolvimento de suas atividades, e as ferramentas de infraestrutura vão dar suporte para a manutenção dessas ferramentas.

A externalização na fase de integração vai colaborar com as documentações e

registros nos relatórios dos erros e acertos existentes nos códigos. As práticas e ferramentas são representadas no Quadro 15.

Externalização na Integração	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH; 	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de infraestrutura.

Quadro 15 – Práticas e ferramentas recomendadas para a externalização na integração
Fonte: Autoria própria.

As políticas de RH vão estar presentes também nessa conversão com os mesmos objetivos, já que na externalização o foco vai ser em relatar os atendimentos nos requisitos.

Para isso, as ferramentas que serão utilizadas serão as ferramentas de captura, que permitirão o registro, e as ferramentas de infraestrutura, que facilitarão as tarefas.

Combinação na Integração
Ferramentas
<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura.

Quadro 16 – Ferramentas recomendadas para a combinação na integração
Fonte: Autoria própria.

A combinação na fase de integração, assim como no desenvolvimento, ocorre somente na base de dados da empresa, existindo assim somente ferramentas para facilitarem tal combinação, vistos no Quadro 16.

As ferramentas encontradas são as de evolução da informação e de infraestrutura, que vão estruturar os conhecimentos específicos e qualificá-los para melhores identificações e subseqüentes usos.

Já no Quadro 17, serão apontadas as práticas e ferramentas para internalização do

conhecimento no processo de integração.

Internalização na Integração	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH; • Participação em congressos e seminários. 	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura.

Quadro 17 – Práticas e ferramentas recomendadas para a internalização na integração
Fonte: Autoria própria.

As políticas de RH vão dar um apoio para os indivíduos buscarem entender todos os códigos e todos os requisitos dos clientes.

Já a participação em congressos e seminários vai permitir a aplicação de conhecimentos para entendimento de códigos, métodos de testes e formas de interpretação do sistema.

As ferramentas de informação vão dar total apoio para entendimento dos códigos, da documentação e do documento de requisitos do cliente. Com isso, ficará mais fácil as interpretações, e a infraestrutura também se fará presente para dar base.

3.6 Gestão do conhecimento na manutenção

Ao final do processo de desenvolvimento do sistema, ou final dessa fase de desenvolvimento, os relatórios de testes precisam ser internalizados para ajustes necessários.

O sistema como um todo precisará ser internalizado pelo cliente para a operação, além de ser necessário socializar as experiências com os clientes para um melhor entendimento.

Além disso, são socializadas as experiências dos clientes na operação dos sistemas,

reportando seus erros e acontecimentos.

Com isso, é possível externalizar os conhecimentos gerados nesses processos para experiências gerais, enviadas para a base de dados, além de externalizar novos requisitos com base nas experiências, testes e ajustes feitos.

Também é necessário externalizar o conhecimento para manuais e tutoriais de usos do sistema como um todo para o cliente aprender a usá-lo.

É possível combinar os conhecimentos gerais do projeto, dos clientes e do sistema na base de dados, mas é uma operação que exige muito mais de máquinas e sistemas do que de pessoas.

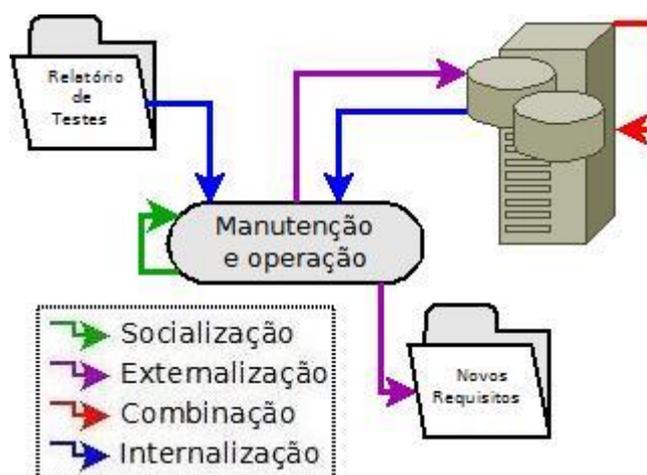


Figura 8 - Conversões do conhecimento na manutenção.
Fonte: Autoria própria.

O documento de Novos Requisitos possui todos os registros realizados no processo de manutenção, com correções e melhorias necessárias para o sistema, permitindo que a equipe possa melhorar e aprimorar o sistema.

Pela Figura 8, vê-se como ficam representadas essas conversões na última fase do processo de desenvolvimento em cascata.

3.6.1 Práticas e Ferramentas de GC na manutenção

A socialização do conhecimento na fase de manutenção é muito mais focada nas experiências dos clientes para entendimento de quais são os problemas ou recomendações. Com isso, no Quadro 18, identificam-se as seguintes práticas e ferramentas.

Socialização na Manutenção	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Políticas de RH; • Geração de ideias; • Participação em congressos e seminários. 	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura.

Quadro 18 – Práticas e ferramentas recomendadas para a socialização na manutenção
Fonte: Autoria própria.

A conectividade entre as pessoas é direcionada para a troca de experiências de indivíduos distintos envolvidos. Também tem de existir a necessidade da conectividade entre as pessoas no relacionamento cliente-empresa.

As políticas de RH e geração de ideias vão permitir o entendimento de que novos projetos podem ser desenvolvidos a partir do que acaba de ser criado.

Os congressos e seminários serão aplicados pela própria empresa que desenvolveu o sistema para apresentar as experiências e ensinar o cliente usar o sistema.

As ferramentas de socialização vão facilitar a comunicação cliente-empresa, permitindo as notificações de erros e não atendimentos. E as ferramentas de infraestrutura manterão as demais ferramentas.

Na externalização de novos requisitos e de experiências, verificam-se as seguintes práticas e ferramentas (Quadro 19).

Externalização na Integração	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH; 	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de infraestrutura.

Quadro 19 – Práticas e ferramentas recomendadas para a externalização na integração
Fonte: A autoria própria.

As políticas de RH vão manter o direcionamento da equipe para o registro dos acontecimentos e as ferramentas de captura darão um apoio para esses registros. As ferramentas de infraestrutura irão dar suporte para tais registros e armazenagens.

Já para combinar tais conhecimentos, são recomendadas apenas ferramentas, já que são processos internos na base de dados. Pelo Quadro 20, temos que as ferramentas de evolução vão colaborar com o desenvolvimento e qualificação dos conhecimentos armazenados, já as ferramentas de infraestrutura darão suporte.

Combinação na Manutenção
Ferramentas
<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura.

Quadro 20 – Ferramentas recomendadas para a combinação na manutenção
Fonte: A autoria própria.

Ao fim, na internalização das informações e dos conhecimentos, as práticas e ferramentas mais recomendadas são representadas no Quadro 21.

A internalização será motivada também pelas políticas de RH, e a participação dos clientes em congressos dados pela empresa desenvolvedora permitirão um entendimento melhor do sistema.

Internalização na Manutenção	
Práticas	Ferramentas
<ul style="list-style-type: none"> • Políticas de RH; • Participação em congressos e seminários. 	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura.

Quadro 21 – Práticas e ferramentas recomendadas para a internalização na manutenção
Fonte: Autoria própria.

As ferramentas de informação serão muito úteis para representações de modelos e construção de manuais para apresentações aos clientes. Já as ferramentas de infraestrutura estarão como base, como em todos processos, para as demais ferramentas.

3.7 Análises e modelos das conversões de conhecimento

É possível verificar algumas conversões que foram comuns em todos os processos de desenvolvimento de software. Com isso, pode-se criar uma representação para o processo de desenvolvimento de software como um todo, permitindo simplificar as representações.

A partir disso, serão criados duas propostas, uma voltado para as conversões em comum e outra para as conversões exclusivas dos processos. Na Figura 9, é visto como se pode representar as conversões em comum.

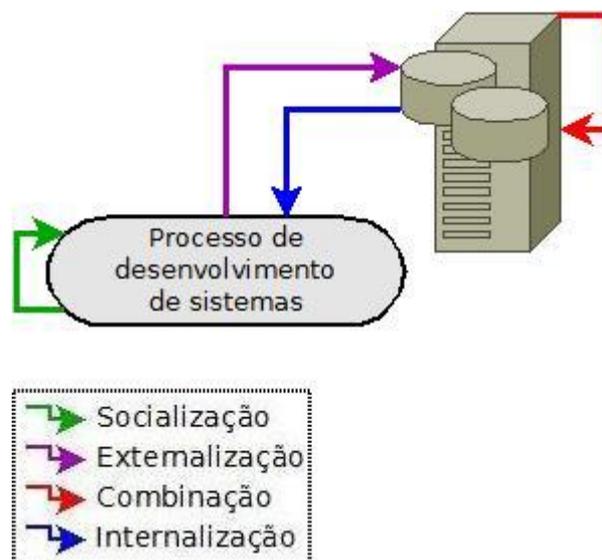


Figura 9 – Proposta de conversões do conhecimento em comum no processo de desenvolvimento de sistemas.

Fonte: Autoria própria.

A socialização é comum em todos os processos que objetiva a integração dos indivíduos presentes em todos os processos.

Com a base de dados, é necessário externalizar conhecimentos de cada fase dos processos para tal base, assim como é possível internalizar conhecimentos que sejam necessários.

Nessa base de dados, pode-se também combinar os conhecimentos existentes, podendo desenvolver melhor os conhecimentos.

Pela Figura 9, verificam-se as conversões de conhecimentos exclusivas de cada processo, sendo algumas conversões comuns dos processos, como as externalizações nos fins dos processos e as internalizações nos começos dos processos, entretanto as elas serão tratadas como exclusivas por se tratar de objetivos diferentes para tais conversões.

Como já dito, é existente uma externalização sempre ao fim de cada processo, assim como uma internalização também. Isso é uma característica do modelo em cascata, ou qualquer outro modelo que trate de documentar os processos executados, pois essa característica fará com que o conhecimento seja

externalizado em um documento no processo anterior, para depois ser internalizado em um processo seguinte.

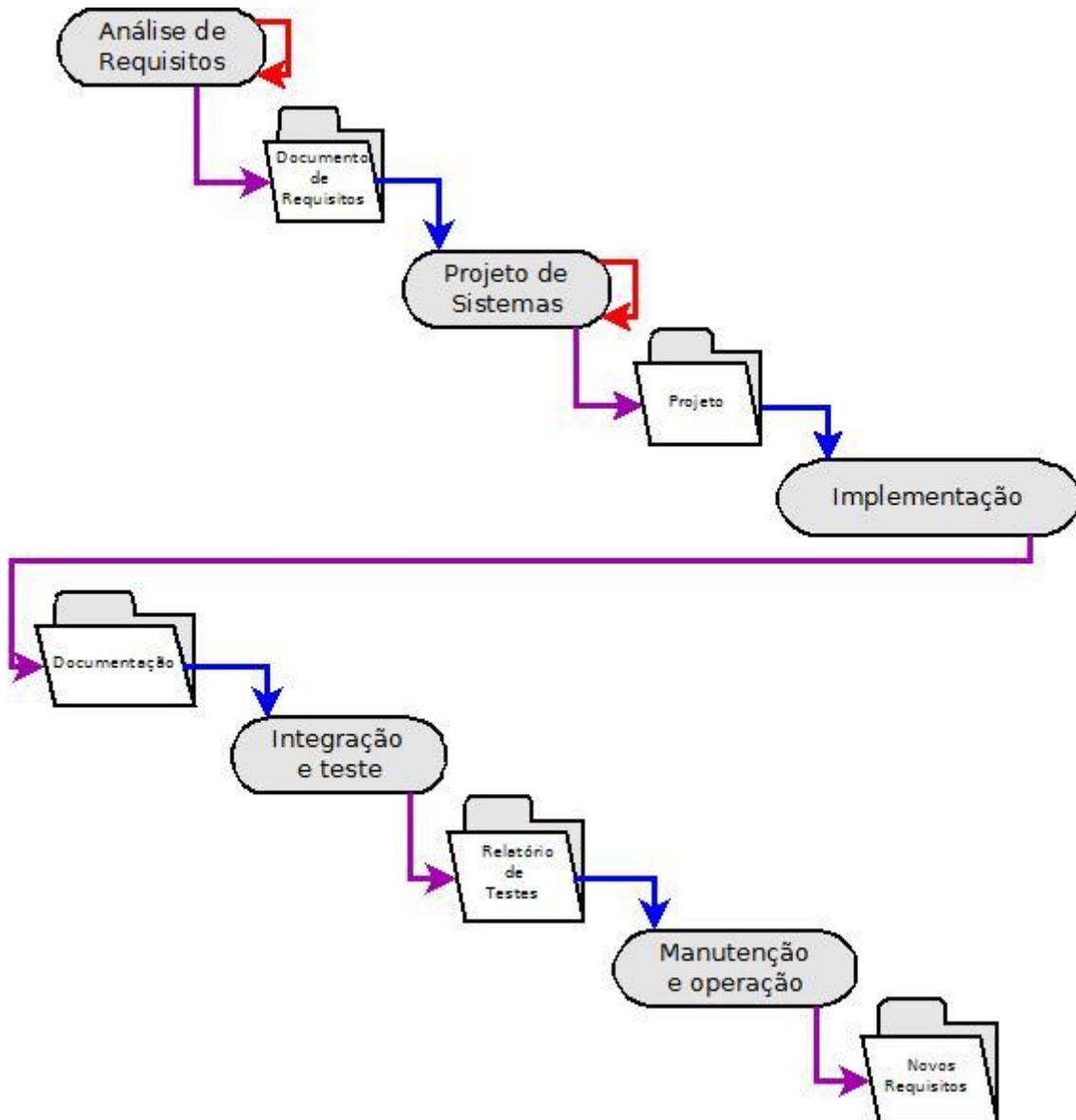


Figura 10 – Modelo de conversões do conhecimento exclusivos de cada processo.
Fonte: Autoria própria.

Essa documentação é característica de equipes independentes e separadas, pois esse trabalho sendo integrado e mantido em um mesmo ambiente, possibilita que tais conversões não fossem necessárias com tanta frequência.

Outro ponto que se pode verificar é que no processo de análise de requisitos e projeto de sistemas existe a combinação de conhecimentos. Isso ocorre pelo objetivo desses processos iniciais, que é exatamente transformar conhecimentos,

tácitos e explícitos, em estruturações lógicas.

A análise de requisitos se faz necessária para construção de requisitos que possam ser entendíveis, assim como é possível identificar requisitos novos com tal combinação.

No projeto de sistemas, os conhecimentos combinados são os requisitos elicitados com os conhecimentos explícitos de modelos e projetos já elaborados. Com isso, é possível reaproveitar muitos conhecimentos, permitindo a construção de um projeto mais eficiente e mais objetivo. Isso pode deixar essa fase muito mais rápida.

Entre os processos de análise de requisitos e projetos de sistemas, as abordagens são individuais, fazendo com que os conhecimentos tenham mais utilidades para os indivíduos. Entre o projeto de sistemas e a implementação já entra uma abordagem de conversões para o grupo. Na integração além do grupo, tem-se uma abordagem de organização, pois o sistema após essa fase é considerado pronto, e todo o conhecimento inerente ao processo todo anterior é de interesse da empresa. Na manutenção, as abordagens retornam para os indivíduos para ajustes e melhorias.

3.8 Análises das práticas e ferramentas

Para uma melhor interpretação, são reunidas todas as práticas e ferramentas citadas nas conversões de conhecimento nos processos. As práticas e ferramentas são apresentadas no Quadro 22.

Práticas e Ferramentas de Gestão do Conhecimento para o Processo de Desenvolvimento de Software			
Processos	Conversões	Ferramentas	Práticas
Análise de Requisitos	Socialização	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura 	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Políticas de RH; • Geração de ideias.
	Externalização	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de evolução; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH.
	Combinação	<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Participação em congressos e seminários; • Políticas de RH
	Internalização	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Trabalho conjunto à diretoria; • Políticas de RH.
Projeto de Sistemas	Socialização	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Políticas de RH; • Geração de ideias; • Mapa das habilidades e competências dos funcionários.
	Externalização	<ul style="list-style-type: none"> • Ferramentas de captação; • Ferramentas de evolução; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH.
	Combinação	<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Participação em congressos e seminários; • Políticas de RH
	Internalização	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH

Práticas e Ferramentas de Gestão do Conhecimento para o Processo de Desenvolvimento de Software			
Implementação	Socialização	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Aperfeiçoamento do layout dos escritórios; • Políticas de RH; • Geração de ideias; • Programa de mentores.
	Externalização	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH; • Participação em congressos e seminários.
	Combinação	<ul style="list-style-type: none"> • Ferramentas de evolução • Ferramentas de infraestrutura 	<ul style="list-style-type: none"> • ---
	Internalização	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH.
Integração	Socialização	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Aperfeiçoamento do layout dos escritórios; • Políticas de RH; • Geração de ideias; • Programa de mentores.
	Externalização	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH.
	Combinação	<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • ---
	Internalização	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH; • Participação em congressos e seminários.

Práticas e Ferramentas de Gestão do Conhecimento para o Processo de Desenvolvimento de Software			
Manutenção	Socialização	<ul style="list-style-type: none"> • Ferramentas de socialização; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Políticas de RH; • Geração de ideias; • Participação em congressos e seminários.
	Externalização	<ul style="list-style-type: none"> • Ferramentas de captura; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH.
	Combinação	<ul style="list-style-type: none"> • Ferramentas de evolução; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • ---
	Internalização	<ul style="list-style-type: none"> • Ferramentas de informação; • Ferramentas de infraestrutura. 	<ul style="list-style-type: none"> • Políticas de RH; • Participação em congressos e seminários.

Quadro 22 – Práticas e ferramentas recomendadas para a todos processos de desenvolvimento de software.
Fonte: Autoria própria.

Com todas as práticas e ferramentas reunidas, é possível inferir sobre algumas práticas e ferramentas.

Praticar o incentivo através de políticas de RH é indicado em todos os processos de desenvolvimento, e em mais de uma conversão por processo. Isso representa a importância que os indivíduos do processo possuem, pois em quase todas as conversões, é necessária a colaboração das pessoas.

Os incentivos e políticas de RH não devem ser direcionados financeiramente apenas, mas devem ser presentes na cultura da empresa e em seus valores, pois muito se tem dito sobre “inovar nas empresas” ou “formadores de conhecimento” dentro das organizações de tal modo que deve ser realmente levado a sério.

A geração de ideias geralmente está vinculada às práticas e políticas de RH, sendo

em muitos dos casos uma consequência de tais políticas.

A participação em congressos e seminários vão auxiliar a empresa, nas conversões de externalização e internalização, já que o processo de aprendizado nos seminários são da forma tácita-implícita para explícita. Os indivíduos aprendem a teoria, internalizando-a, mas precisarão colocá-las em prática. A internalização também é beneficiada na fase de integração, onde os aprendizados nos congressos farão com que as interpretações dos códigos sejam feitas de forma mais eficientes.

O apoio dos congressos e seminários para a externalização tem o enfoque nas práticas de desenvolvimento, seja de modelos e gráficos ou de construção de códigos. Isso se dá pelo fato da prática, no desenvolvimento de software de um conhecimento explícito, ocorrer pelas tentativas de externalizá-lo.

As práticas para a socialização tem um enfoque mais no layout dos escritórios e na conectividade entre as pessoas. Criando um canal de comunicação mais eficiente, o conhecimento será compartilhado mais eficientemente também.

Com um enfoque nas ferramentas, é mais clara uma divisão de ferramentas por tipo de conversão de conhecimento. Mas um tipo de ferramenta é notado como fundamental em todas conversões. As ferramentas de infraestrutura.

Tais ferramentas de podem ser consideradas como a base para a manutenção das demais ferramentas de gestão do conhecimento. Isso não indica que os investimentos maiores precisem estar na infraestrutura, mas é necessário investir o mínimo para que as demais ferramentas possam existir.

Considerando as conversões de conhecimento, na socialização as ferramentas que mais apresentam apoio são as de socialização que facilitarão o fluxo de informações e a conectividade entre as pessoas.

Para a externalização, as ferramentas de captura foram destacadas, pois essas dariam total apoio para os indivíduos desenvolverem seus conhecimentos nelas.

As ferramentas de evolução do conhecimento apareceram em todas as conversões de combinação de conhecimento. Sendo essas ferramentas próprias para isso, elas não requerem uma operacionalização de todos constantemente. Mas seus resultados podem ser diferenciais na gestão do conhecimento.

E, por fim, na internalização, as ferramentas de informação dão total apoio para disponibilizar essas informações no momento certo, na forma correta e no local correto.

No Quadro 23, apresentam-se as práticas específicas e diferenciadas para cada processo de desenvolvimento de software. As análises serão feitas generalizadas para todas as conversões.

Tais práticas são específicas conforme a característica de cada processo de desenvolvimento.

Na Análise de Requisitos, tal processo tem como objetivo captar os conhecimentos de dentro da empresa e do cliente, sendo o relacionamento entre as duas partes um fator determinante para a eficácia do processo.

Sendo assim, percebe-se a prática de “conectividade entre as pessoas” presente para as conversões de conhecimentos, analisando-se que tais pessoas seriam os clientes e os responsáveis na coleta de requisitos. A “participação em congressos e seminários” vai colaborar com a melhoria na captação dos requisitos dos responsáveis. E o “trabalho em conjunto à diretoria” permite que muitos requisitos, que estão na cabeça dos diretores das empresas, e que, em muitos momentos, são os responsáveis pela coleta de requisitos, sejam elicitados.

Práticas específicas de Gestão do Conhecimento	
Processo de Desenvolvimento	Práticas Específicas
Análise de Requisitos	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Participação em congressos e seminários; • Trabalho conjunto à diretoria.
Projeto de Sistemas	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Mapa das habilidades e competências dos funcionários; • Participação em congressos e seminários.
Implementação	<ul style="list-style-type: none"> • Aperfeiçoamento do layout dos escritórios; • Programa de mentores; • Participação em congressos e seminários.
Integração	<ul style="list-style-type: none"> • Aperfeiçoamento do layout dos escritórios; • Programa de mentores; • Participação em congressos e seminários.
Manutenção	<ul style="list-style-type: none"> • Conectividade entre as pessoas; • Participação em congressos e seminários.

Quadro 23 – Práticas específicas de gestão do Conhecimento em cada processo de desenvolvimento de software
Fonte: Autoria própria.

Na fase de Projeto de Sistemas ainda é necessária a “conectividade entre as pessoas” para a estruturação dos requisitos, além de abstrair os conhecimentos dos responsáveis pela análise de requisitos. Nesse processo, obter um “mapa das habilidades e competências dos funcionários” vai permitir uma melhor mensuração de prazos, custos e riscos da empresa. E por fim “participação em congressos e seminários” vai permitir um melhor conhecimento para a estruturação dos modelos, diagramas e projetos.

Para a Implementação, tem-se grande foco para a socialização, possibilitada por um

“aperfeiçoamento do layout dos escritórios” e com um “programa de mentores”. Já na “participação em congressos e seminários”, a melhoria ficaria na construção dos códigos e na melhoria das linguagens de programação.

Tendo uma característica muito similar à Implementação, a Integração apresenta as mesmas práticas no que se diz respeito à socialização: “aperfeiçoamento do layout dos escritórios” e “programa de mentores”. A diferença se mostra na “participação em congressos e seminários”, onde anteriormente o enfoque era na externalização do conhecimento, e nessa fase o enfoque é na internalização do conhecimento, ou seja, na leitura dos códigos e aprendizado do que foi desenvolvido.

A Manutenção se apresenta muito similar a Análise de Requisitos, mas com um sistema pronto e desenvolvido. A “conectividade entre as pessoas” será crucial para o alinhamento do cliente com a equipe de desenvolvimento. E, por fim, a “participação em congressos e seminários” poderá permitir ao cliente aprender sobre o sistema, e à equipe aprender sobre novas tecnologias.

4 CONSIDERAÇÕES FINAIS

4.1 Contribuição

Com esse trabalho, pode-se apresentar um modelo de como o conhecimento se comporta no processo de desenvolvimento de software, uma vez que não foi encontrado um modelo representativo especificamente para esse processo.

Com isso, fica mais fácil para os interessados em aplicar a gestão de conhecimento em seus processos de desenvolvimento, além de terem em vista as ferramentas e práticas que irão facilitar esse trabalho.

Considerando o modelo teórico, ele precisa ser validado em um processo real, e ser verificada qual a efetividade do modelo, das práticas e das ferramentas em questão.

Ainda assim, foi possível iniciar o estudo e a pesquisa nessa área, permitindo que futuros trabalhos possam se aprofundar e melhorar a gestão do conhecimento nas empresas que possuem necessidade.

4.2 Dificuldades e Limitações

Como o trabalho é teórico, mesmo com enfoque na prática, foram necessárias várias simplificações na análise dos processos como um todo.

Não está sendo considerada qual a estrutura real das empresas para a modulação, pois tal consideração requer um estudo muito aprofundado quanto à característica do setor. Em muitos casos, as empresas fazem a alocação de seus funcionários conforme as suas necessidades, sem nenhum preparo ou planejamento. Com isso, o planejamento da gestão do conhecimento dentro da empresa também é esquecido.

A simplicidade do modelo em cascata também foi uma decisão que pode ter limitado alguns pontos, facilitando na modelagem dos processos. Entretanto podendo limitar ou precisar de ajustes na implementação do modelo proposto.

4.3 Trabalhos Futuros

Como um dos princípios desse trabalho, uma das necessidades é a validação do modelo, das ferramentas e das práticas em um processo real, considerando indicadores de desempenho de gestão do conhecimento, possibilitando uma análise mais realista.

Outro ponto é considerar os impactos que as práticas podem trazer para as empresas, além de quais os recursos necessários para a implantação dessas práticas e, com isso, verificar a viabilidade das ferramentas também.

Também é possível desenvolver esse modelo para a aplicação nos demais tipos de desenvolvimento de software, assim como foi feito na modelação do processo em cascata para outros tipos de desenvolvimento. Isso permite a evolução desse modelo para demais processos como o desenvolvimento incremental, desenvolvimento ágil, etc.

REFERÊNCIAS

ANDERSEN, A. **The Knowledge Management Tool**. The American Productivity and Quality Center, Chicago, n.0, p.14, 1997. Disponível em: <<http://www.apqc.org/km/>> Acesso em: 9 jun. 2013

ANDRADE, M. T. T. **Uma proposta para a gerência do conhecimento ao longo do processo de desenvolvimento de produto**. 2005. 186f. Dissertação (Mestrado em Redes de Computadores) - Universidade Salvador, Rio de Janeiro, 2005.

ANQUETIL *et al.* **Software maintenance seen as a knowledge management issue**. Information and Software Technology, Brasília, n.49, p.515 – 529, 2007.

AURUM, A.; DANESHGAR, F.; WARD, J. **Investigating Knowledge Management practices in software development organizations - An Australian experience**. **Investigating Knowledge Management practices in software development organizations - An Australian experience**, School of Information Systems - University of New South Wales. Sydney, n.2052, p.*, 2007.

AURUM, A.; WARD, J. **Knowledge Management in Software Engineering – Describing the Process**. Australian Software Engineering Conference, Sydney, n.4, p.*, 2004.

AVRAM *et al.* **Knowledge Management in Distributed Software Development Teams - Does Culture Matter?**. Fourth IEEE International Conference on Global Software Engineering, Porto, n.10, p.*, 2009.

BASIL, V. R.; OIVO, M. **Representing Software Engineering Models: The TAME Goal Oriented Approach**. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Finlandia, v.18, n.10, p.886 - 897, 1992.

BERTOLINO, A. **Software Testing Research: Achievements, Challenges, Dreams**. Future of Software Engineering, Pisa, Itália, n.5, p.*, 2007.

BOEHM, B. W. **A Spiral Model of Software Development and Enhancement**. IEEE, California - Estados Unidos, n.88, p.61 - 73, 1988.

BOSCOLO *et al.* **PRÁTICAS E FERRAMENTAS UTILIZADAS PARA A GESTÃO DO**. Revista Gestão Industrial, Ponta Grossa, n.5, p.87-104, 2009.

CHEN, H.; HU, P. J.; WEI, C. **Design and Evaluation of a Knowledge**

Management System. IEEE Software, *, v.19, n.3, p.56 - 59, 2002.

COSTA, P. E. C. **Desenvolvimento de um modelo de gestão do conhecimento para a melhoria do processo de desenvolvimento de produto.** 2005. 132f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal do Rio Grande do Norte, Natal - RN, 2005.

DAVENPORT, T. H.; PRUSSAK, L. **Conhecimento Empresarial: Como as organizações gerenciam o seu capital intelectual.** 4. ed. Rio de Janeiro: Campus, 1998. 500p.

DISTERER, G. **Management of project knowledge and experiences.** Journal of Knowledge Management, Alemanha, v.6, n.5, p.512 - 520, 2002.

FEHÉR, P.; GÁRBOR, A. **The Role of Knowledge Management Supporters in Software Development Companies.** Software Process Improvement Practices, Budapest, n.11, p.251-260, 2006.

GUINDON, R. **Knowledge exploited by experts during software system design.** Int. J. Man-Machine Studies, Estados Unidos, v.1, n.33, p.274 - 304, 1990.

HERBSLEB, J. D.; MOITRA, D. **Global Software Development.** IEEE Software, India, v.18, n.2, p.15 - 20, 2001.

HOLZ, H.; MAURER, F. **Process-Oriented Knowledge Management For Learning Software Organizations.** Knowledge-Based Systems Workshop, Chicago, n.99, p.*, 1999.

HOOG, R.; SPEK, R. V. D.; WIIG, K. **Supporting knowledge management: a selection of methods and techniques.** ElsevierScience Ltd, Texas, n.5, p.15 - 27, 1997.

IEEE COMPUTER SOCIETY **SWEBOK.** 2004. ed. California, Estados Unidos: IEEE, 2004. 202p.

KONNO, N.; NONAKA, I. **The Concept of Ba. Building a foundation for knowledge creation.** California Management Review, California, n.40, p.40, 1998.

LAMSWEERDE, A. V. **Requirements Engineering in the Year 00: A Research**

Perspective. ICSE 2000, Limerick, Ireland , n.1, p.5 - 19, 2000.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação gerenciais.** Person Prentice Hall, São Paulo, v.1, n.1, p.*, 2004.

NONAKA, I. **A Empresa Criadora de Conhecimento.** Harward Businesse Review, Rio de Janeiro, n.1, p.*, 1991.

NONAKA, I.; TAKEUCHI, H. **The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation.** Oxford University Press, New York, n.1, p.*, 1995.

NONAKA, I.; TAKEUCHI, H. **Gestão do Conhecimento.** Porto Alegre: Bookman, 2008. 320p.

PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach.** 6. ed. International Edition: McGraw Hill, 2004. 711p.

RAO, M. **KM tools and techniques : practitioners and experts evaluate KM solutions.** 1. ed. Oxford: Elsevier Butterworth–Heinemann, 2005. 453p.

SCHNEIDER, K. **Experience and Knowledge Management in Software Engineering.** Hannover: Springer, 1998. 247p.

SILVA, S. L. **Gestão do conhecimento: uma revisão crítica.** Ci Inf., Brasília, n.33, p.143-151, 2004.

SOMMERVILLE, I. **Engenharia de Software.** 6. ed. São Paulo: Addison Wesley, 2003. 500p.

WERNER, W.; WERNER, I. A. **GESTÃO DO CONHECIMENTO: FERRAMENTAS TECNOLÓGICAS E PORTAIS DO CONHECIMENTO PARA EMPRESAS DESENVOLVEDORAS DE TECNOLOGIAS DE MÉDIO E PEQUENO PORTES.** TERRA E CULTURA, Londrina, n.38, p.182-240, 2004. Disponível em: <http://web.unifil.br/docs/revista_eletronica/terra_cultura/38/Terra%20e%20Cultura_38.pdf> Acesso em: 10 jun. 2013

Knowledge Management.. Site resultante da Tese de Mestrado de Alan Frost (2010) Disponível em: <<http://www.knowledge-management-tools.net/>> Acesso em:

13 ago. 2013