UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CLAUDIONOR SANTOS DE OLIVEIRA
RONALDO JOSÉ ABEL

BOLACHA : ANÁLISE DE FERRAMENTAS PARA GERÊNCIA DE PROJETOS EM SOFTWARE LIVRE.

FLORIANÓPOLIS
2011

CLAUDIONOR SANTOS DE OLIVEIRA
RONALDO JOSÉ ABEL

BOLACHA : ANÁLISE DE FERRAMENTAS PARA GERÊNCIA DE PROJETOS EM SOFTWARE LIVRE.

Trabalho apresentado ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina como requisito para obtenção do título em Bacharel em Sistema de Informação.

Orientador: José Eduardo de Lucca.

FLORIANÓPOLIS
2011

CLAUDIONOR SANTOS DE OLIVEIRA
RONALDO JOSÉ ABEL

BOLACHA : ANÁLISE DE FERRAMENTAS PARA GERÊNCIA DE PROJETOS EM SOFTWARE LIVRE.

Orientador:

_____
José Eduardo De Lucca (INE – CTC – UFSC)

Banca:

_____
Christiane Gresse von Wangenheim (INE – CTC – UFSC)

_____
Yuri Gomes Cardenas (GeNESS - UFSC)

RESUMO

Atualmente a crescente demanda de novos projetos de software tem ressaltado diversos problemas em seu processo de desenvolvimento. Em particular, o Software Livre, pois se difere do modelo tradicional de desenvolvimento de um software. Para auxiliar neste cenário, a adoção de ferramentas que fornecem suporte a gerência de projetos pode conduzir a um software com maior grau de qualidade. Dentro desse contexto, é realizada uma análise comparativa entre ferramentas para gerência de projetos, correlacionando os seus recursos com os utilizados no desenvolvimento de softwares livres, segundo a revisão bibliográfica realizada no trabalho.

Palavras-chave: Software Livre. Gerência de Projetos. Ferramentas gerência de projetos.

ABSTRACT

Currently the growing demand for new software projects, has jutting out several problems in its development process. In particular, the Free Software, because it differs from the traditional model of software development. However, the adoption of programs for support project management, can lead to a software with higher quality. within this context, an analysis was applied between the programs for support project management, correlating with resources used in development of free software, according to the literature review in this study.

Keywords: Free Software. Project Management. Project Management Software.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

ASQ	American Society for Quality
ANSI	American National Standard
CMMI	Capability Maturity Model Integration
CRUD	Create, Read, Update, Delete
CVS	Concurrent Versions System
DFSG	Debian Free Software Guidelines
EC	European Commission
ERP	Enterprise Resource Planning
FLOSS	Free, Libre, Open Source, Software
FSF	Free Software Fundation
GPL	General Public License
GNU	GNU is Not Unix
IBM	International Business Machines
ICT	Information and communication technologies
IDE	Integrated development environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Organization for Standardization
MVC	Model,View,Controller
MPS.BR	Melhoria de Processos do Software Brasileiro
PDCA	Plan, Do, Check e Action
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
OSI	Open Source Initiative
SGBD	Sistema Gestor de Base de Dados
SOFTEX	Sociedade Brasileira para Promoção da Exportaçao de Software
SL	Software Livre
SPI	Software in the Public Interest
SVN	Subversion
SWEBoK	Software Engineering Body of Knowledge
TI	Tecnologia da Informação

SUMÁRIO

1. INTRODUÇÃO

Neste capítulo é apresentado o estudo realizado neste trabalho, discutindo a contextualização e justificativa, os objetivos gerais, objetivos específicos do trabalho e a estruturação do trabalho.

1.1 CONTEXTUALIZAÇÃO E JUSTIFICATIVA

Segundo a empresa Sun Microsystems (2009), o software livre ao longo dos anos tem crescido com o mesmo potencial que possui para ajudar tanto as empresas, como instituições e indivíduos, os quais buscam ferramentas que auxiliem em seus processos internos. De acordo com AGUIAR et al. (2009), um software livre é definido pela sua forma de licenciamento, ou seja, o modo como é regulamentada.

Essa demanda crescente de desenvolvimento de software livre vem seguida de uma necessidade de planejamento para atender aos objetivos de quem procura ou requisita o software. De acordo BROD e KÄFER (2009), muitas vezes, não existe um planejamento inicial, ou seja, a ausência de procedimentos definidos de coordenação, além da ausência de uma gerência de configuração que garanta a qualidade do software. Segundo a *American Society for Quality* (2010, apud PMBOK, 2004, p. 194), podemos considerar qualidade de software, como o grau até o qual um conjunto de características inerentes satisfaz as necessidades e, de acordo com DIAS (2011), a Gerência de Configuração é um conjunto de atividades que permita o controle das mudanças em um processo de desenvolvimento de software.

Segundo o PMBOK (2008), uma metodologia, define as abordagens ferramentas e fontes de dados que podem ser usadas para realizar o gerenciamento. Sendo que as metodologias de gerência de projetos existentes no mercado atualmente utilizam técnicas na elaboração das atividades para atingir aos objetivos, em um prazo, com um custo determinado e garantias mínimas de qualidade.

De acordo com MARJORIE e PERREIRA (2006), os processos de desenvolvimento de um software livre possuem padrões que diferem dos modelos tradicionais, não havendo um único gestor responsável pelo planejamento inicial, mas sim um grupo de pessoas, as quais possuem diferentes papéis em cada fase do projeto.

Portanto, o grande desafio para a proposta deste trabalho é fundir os conceitos das metodologias de gerência de projetos com a visão do desenvolvimento em software livre, diante de suas características peculiares. Conforme BROD e KÄFER (2009), os seguintes aspectos se diferem no desenvolvimento do software livre:

a) O processo de desenvolvimento de um software é realizado por centenas ou milhares de voluntários, ainda que alguns destes voluntários tenham seu trabalho patrocinado por suas empresas;
b) O trabalho não é designado, os desenvolvedores ou usuários escolhem as tarefas que querem executar;
c) Grande parte dos projetos surge de motivações pessoais e replicam funcionalidades de algum projeto existente.

Este trabalho possui grande importância para o software livre, uma vez que o estudo permite identificar as principais características no seu modelo de desenvolvimento de forma comparativa com as ferramentas de gerência de projetos existente no momento. Uma ferramenta de gerência de projetos consiste da agregação dos processos, ferramentas, técnicas, metodologias, recursos e procedimentos para o gerenciamento de um projeto (PMBOK, 2008, p.322).

1.2 OBJETIVO

1.2.1 Objetivo Geral

Realizar uma análise comparativa sobre um grupo de ferramentas para gerência de projetos, de tal modo que seja possível correlacioná-las com as características identificadas do modelo de desenvolvimento de um software livre.

1.2.2 Objetivo Específicos

- Levantar as características do processo de desenvolvimento do software livre;
- Levantar as características das metodologias de gerência de projetos;
- Levantar ferramentas de suporte à gerência de projetos.

1.3 ESTRUTURA DO TRABALHO

O primeiro capítulo apresenta uma introdução sobre o trabalho, contextualizando o tema adotado de acordo com os modelos existentes e suas respectivas referências, apresentando a sua importância, além da definição do objetivo principal e as etapas de desenvolvimento do trabalho.

O segundo capítulo apresenta uma revisão bibliográfica sobre metodologias de Gerência de Projetos, seus conceitos, modelos e as principais características identificadas.

O terceiro capítulo apresenta uma revisão bibliográfica sobre o software livre e os processos de desenvolvimento de software, além de seus princípios e características.

O quarto capítulo apresenta um conjunto de ferramentas para a Gerência de Projetos, selecionadas de acordo com as características definidas ao longo do trabalho, realizando uma análise comparativa com os padrões e princípios de desenvolvimento do Software Livre, apresentado no terceiro capítulo.

Finalmente, o último capítulo apresenta uma conclusão sobre o tema abordado.

2. GERÊNCIA DE PROJETOS

Neste capitulo é apresentado uma revisão bibliográfica sobre os principais conceitos que abrangem a Gerência de Projetos, metodologias, modelos e as principais características identificadas.

2.1 CONCEITOS

Antes de iniciar o estudo, é importante ressaltar o significado de um projeto. De acordo com o PMBOK (2008), o projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo, com uma natureza temporária e bem definida, com uma data de início e fim pré-definidas, ou até que os objetivos propostos sejam alcançados.

Desta forma, a gerência de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos (PMBOK, 2008) ou objetivos pré-definidos, num certo prazo, com um custo estabelecido e metas de qualidades a serem cumpridas, através da mobilização de recursos financeiros, tecnológicos e humanos.

Tabela 1. Diferenças entre um projeto e uma operação contínua.

Projeto	Operação continua
Temporário: tem um começo e um fim definidos.	Repetitiva: o mesmo processo é repetido várias vezes.
Produz um resultado ou produto único.	Objetiva produzir os mesmos resultados cada vez que o processo é executado.

Fonte: ARANTES et al , 2008, p. 6.

Segundo ARANTES et al. (2008), por ter, obrigatoriamente, início e fim definidos, os projetos se diferenciam de operações contínuas. Essa característica implica que são iniciados, durante seu processo, evoluem e, por fim, são finalizados.

Essas temporalidades dos projetos exigem a adoção de um ciclo de vida, similar ao ciclo de vida de um software, constituído de fases elaboradas com o planejamento das atividades a serem realizadas.

De acordo com o PMBOK (2008), os processos relacionados às suas atividades a serem realizadas durante a execução do projeto podem ser inseridos em grupos, e difundidos em áreas de conhecimento.

Segundo o PMBOK (2008), os grupos de processo podem ser definidos da seguinte maneira:

- **Iniciação** - É a fase inicial de todo projeto, onde ao final da fase é obtido um documento formal representando o projeto e seu objetivo. É uma fase importante para definição do projeto em geral e identificação das partes interessadas, reconhecendo de uma forma direta ou indireta seus objetivos, dando sentido para a execução do projeto.
- **Planejamento** - As ações, entradas e saídas dessa fase são necessárias para definir e executar os pacotes de trabalho necessários para conclusão do projeto e obtenção do resultado. Os pacotes de trabalhos definidos nessa fase do projeto são extremamente importantes para desenvolver o curso do projeto. Ou seja, é a fase responsável por detalhar tudo que será realizado, incluindo prazo, custos, atividades, recursos financeiros e humanos, etc. Outros planos também podem ser desenvolvidos nessa fase, entre eles, comunicação, qualidade, riscos e suprimentos.
- **Execução** - Analisando do ponto de vista do desenvolvimento de um software tradicional, a fase de execução pode ser considerada a fase de implementação, produção e materialização do que foi planejado nas fases anteriores. O PMBOK define essa fase como processos que executam o trabalho definido no planejamento satisfazendo os requisitos do projeto.
- **Monitoramento e controle** - Esta fase não necessariamente ocorre somente em paralelo com a fase de Execução, mas deve acontecer durante a fase de Planejamento e preferencialmente entre as demais fases. Pois é preciso acompanhar, rever e ajustar o progresso e desempenho do projeto, possibilitando que sejam identificadas as áreas que necessitam de mudanças e ajustes, tomando as ações corretivas para ajustar as atividades do projeto em função do objetivo do projeto.

- **Encerramento** - De acordo com o PMBOK, essa última fase contém uma importância para todo o projeto, pois ocorrem a avaliação e homologação das atividades executadas e pacotes de trabalhos entregues, além da atualização da documentação e aprendizados ocorridos durante as fases do projeto. As falhas e problemas ocorridos são base para o aprendizado do projeto, e definição das melhores estratégias para projetos futuros. No entanto, sua principal função é formalizar a aceitação ou encerramento do projeto.

Pelas interações entre seus grupos, os processos de gerenciamento de projetos se desdobram em várias áreas de conhecimento interligadas e interdependentes (ARANTES et al., 2008, p.12). Segundo o PMBOK, uma área de conhecimento é definida por seus requisitos de conhecimento e em termos que a compõem entre outros fatores. O PMI organiza os processos em nove áreas de conhecimento, caracterizam os principais aspectos envolvidos em um projeto e no seu gerenciamento:

- **Gerenciamento de Integração** - Descreve os processos que garantem que os elementos do projeto estão apropriadamente coordenados.
- **Gerenciamento de Escopo** - Descreve os processos necessários para garantir que o projeto inclui todo o trabalho requerido, e somente o trabalho requerido, para que seja completado com sucesso.
- **Gerenciamento do Tempo** - Descreve os processos que garantem que o projeto seja concluído no tempo esperado.
- **Gerenciamento de Custo** - Descreve os processos necessários para garantir que o projeto seja completado dentro do orçamento previsto.
- **Gerenciamento da Qualidade** - Descreve os processos necessários para que o projeto satisfaça as necessidades para o qual se criou.
- **Gerenciamento de Recursos Humanos** - Descreve os processos para garantir o planejamento adequado da equipe e o uso mais eficiente das pessoas envolvidas no projeto.
- **Gerenciamento de Comunicações** - Descreve os processos necessários para que a informação do projeto seja gerada, coletada, disseminada, armazenada e/ou descartada da forma correta.

- **Gerenciamento de Risco** - Descreve os processos que identificam, analisam e respondem aos riscos do projeto. Através da identificação de riscos, quantificação e qualificação de riscos.
- **Gerenciamento de Aquisições** - Descreve os processos necessários para a aquisição de bens e serviços de terceiros.

Na sequência, Figura 1 com as nove áreas de conhecimento e seus respectivos processos.

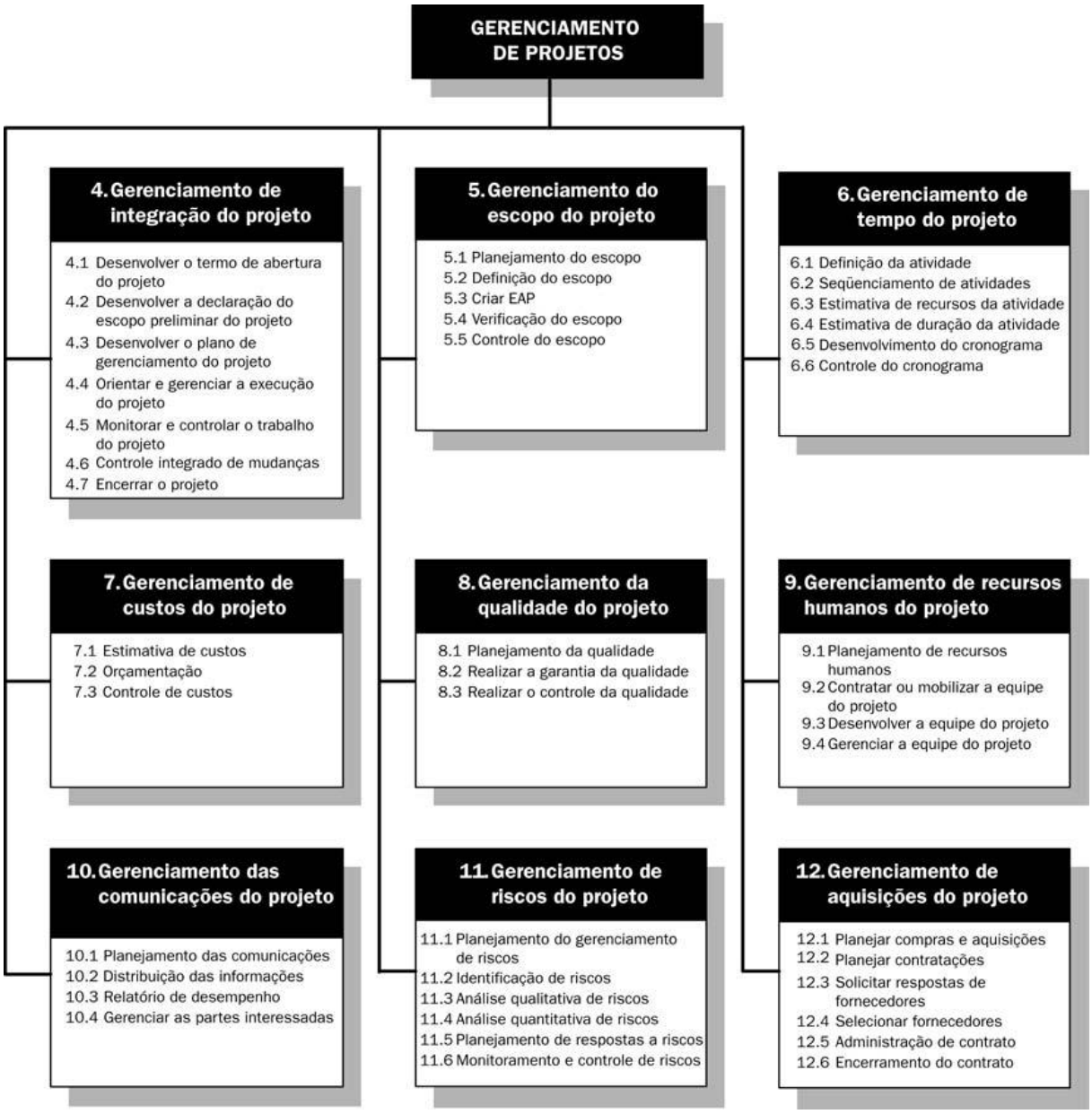

Figura 1 - Visão geral das áreas de conhecimento em gerenciamento de projetos e os 44 processos.
Fonte: PMBOK, 2004, p.11.

2.2 METODOLOGIAS DE GERENCIAMENTO DE PROJETOS

De acordo com SAMPAIO (2008), uma metodologia é composta por técnicas e processos que visam a aumentar e garantir a eficiência do trabalho realizado dentro de uma organização, sendo um fator crítico em gerência de projetos. Segundo BARBI (2010), uma metodologia permite controlar melhor o processo e, com sua utilização, a equipe será mais eficiente, pois entregará o projeto com maior grau de acerto em termos de prazos e custos.

COCKBURN (2001, apud OLIVEIRA 2008) afirma que as metodologias na área de desenvolvimento de software possuem uma estrutura comum entre elas, nas quais listamos abaixo:

- **Papéis** - Pessoa responsável por uma função, com s habilidades necessárias.
- **Habilidades** - As competências necessárias para cada papel.
- **Equipes -** Os papéis que trabalham em grupo para alcança um objetivo.
- **Técnicas** - Os procedimentos ou habilidades necessários para o cumprimento das atividades.
- **Atividades** - Esforço temporário empregado por uma pessoa para o cumprimento de uma meta.
- **Processos** - Como as atividades interagem entre si, com possíveis pré e pós-condições entre atividades.
- **Produto** - O resultado ou objetivo alcançado de um projeto. Podendo, ter um tempo de vida curto, ou mais longo que o próprio projeto.
- **Marcos** - Eventos que marcam o progresso ou fim das atividades.
- **Padrões** - As convenções que uma organização adota no seu processo de desenvolvimento.
- **Valores da equipe** - Fatores culturais, regionais ou aspectos da organização que influência no modo em que a equipe realiza o seu trabalho.

De maneira geral, esses modelos apresentam metas ou estruturas necessárias para que um processo de desenvolvimento apresente melhorias na

qualidade de seu produto, ou seja, melhorias necessárias no processo de desenvolvimento de um software (TAMAKI, 2007, apud CAMPOS e LIMA, 2008). Entre os principais padrões e modelos propostos, podemos dividi-los em metodologias de desenvolvimento ágil, como o SCRUM, ou tradicional, como PMBOK.

Tabela 2: Principais diferenças entre Modelos Ágeis e Modelo Tradicional.

Características	Modelo Ágil	Modelo Tradicional
Premissa Fundamental	Ênfase na Agilidade	Ênfase no Controle Operacional
Condução dos Trabalhos	Baseado em Processos Empíricos	Baseado em Processos Formais
Escopo da Solução	Centradas no Desenvolvimento	Englobam todas as Disciplinas
Profundidade da Abordagem	Definir apenas o quê deve ser feito	Definir o quê e como deve ser feito
Foco dos Profissionais	Atuação Local (por projeto)	Atuação Global (por disciplina)
Abordagem Estratégica	Atender melhor o Curto Prazo	Atender melhor o Longo Prazo
Palavras-Chave	Pessoas, FeedBack, Adaptação	Maturidade, Estrutura, Padronização
Modelos de Implementação	XP, SCRUM	ISO, PMI
Frase que resume sua Filosofia	Aproxime sua equipe do Cliente, simplifique o projeto e aumente sua produtividade	Não podemos melhorar o que não podemos controlar

Fonte: BARTIE, 2007, p. 23. Adaptada.

Entre as metodologias existentes para a Gerência de Projetos, destacamos o PMBOK, por ser corpo de conhecimento da gerência de projeto, além de um padrão de gerência de projetos aprovado pela ANSI (NARDI, 2010). O *Project Management Body Of Knowledge* ou como é mais conhecido, PMBOK[1] é o principal documento de referência do PMI (PMBOK, 2008). Como é definido, ele é um guia do conjunto de conhecimentos em Gerência de Projetos. Está atualmente na quarta edição, lançada em 2008. Como ele mesmo descreve, o seu objetivo é

[1] A sigla PMBOK (Guide to the Project Management Body of Knowledge) é uma marca registrada do Project Management Institute (PMI).

identificar um subconjunto do conhecimento.

> O Guia PMBOK identifica um subconjunto do conjunto de conhecimentos em gerenciamento amplamente reconhecido como boa prática. ‘Amplamente reconhecido’ significa que o conhecimento e as práticas descritas são aplicáveis à maioria dos projetos na maior parte do tempo e que existe um consenso em relação ao seu valor e sua e sua utilidade (PMBOK, 2008, p. 10).

O PMBOK apresenta uma definição de gerenciamento de projetos mais preocupada com sua aplicação prática (ARANTES et al., 2008, p. 9). Dessa forma, podemos relacionar com uma aplicação de conhecimentos, habilidades, ferramentas e técnicas nas atividades a serem realizadas, visando a alcançar os objetivos e expectativas, buscando um ponto de equilíbrio entre as atividades a serem realizadas no início do projeto até sua conclusão.

Além dos padrões que visam o ponto de equilíbrio, o guia estabelece diretrizes para os processos, ferramentas e técnicas de gerenciamento de projetos. Além disso, o Código de Ética e Conduta Profissional do PMI exige que os profissionais que queiram adotar o guia como ferramenta demonstrem um compromisso com a conduta ética e profissional (PMBOK, 2008, p. 11).

De acordo com NARDI (2010), outra metodologia de grande destaque é o SCRUM, recomendada para projetos nas áreas de software e ideal para projetos dinâmicos e suscetíveis a mudanças de requisitos, sejam eles novos ou apenas modificados.

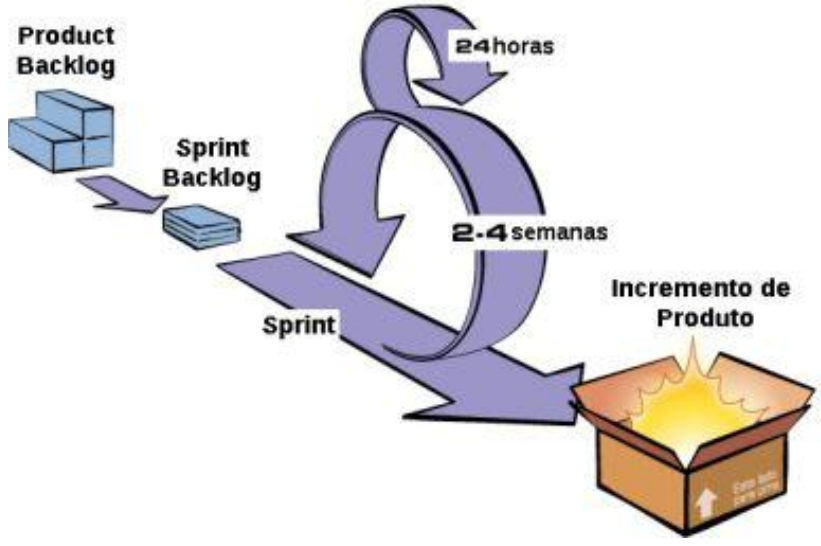

Figura 2. Processos de interações no SCRUM.

Fonte: CISNEIRO et al., 2009 p.10.

Conforme KNIBERG (2007), o SCRUM pode ser definido como uma metodologia, na qual as equipes desenvolvem o projeto em pacotes de trabalho e que esses são entregues ao fim de cada iteração, conforme podemos observar na Figura 2. O SCRUM originalmente foi desenvolvimento para o gerenciamento de projetos de fabricação de automóveis e produtos de consumo, posteriormente adaptado por Ken Schwaber para o desenvolvimento de software, após sua formalização e definição.

Segundo BECK et al.(2001, apud SGANDERLA e LACERDA , 2010), o SCRUM se baseia em 12 princípios, aos quais os métodos ágeis de desenvolvimento de software devem se adequar, esses princípios foram definido pela Aliança Ágil, organização sem fins lucrativos que procura promover o conhecimento e discussão sobre todas as metodologias ágeis (FOWLER, 2003). Estes princípios são:

1. Maior prioridade é satisfazer ao cliente, mediante entregas de software de valor em tempo hábil e continuamente;
2. Aceitar mudanças de requisitos, em qualquer fase do projeto;
3. Entregar software na menor escala possível de tempo;
4. Equipe de desenvolvimento e cliente são do mesmo time;
5. Construir projetos com indivíduos motivados e comprometidos com o resultado;
6. Comunicação efetiva;
7. Software funcionando é a principal medida de progresso;
8. Promover o desenvolvimento sustentável;
9. Atenção contínua à excelência técnica;
10. Simplicidade;
11. As melhores arquiteturas, requisitos e projetos provêm de equipes organizadas;
12. Refletir sobre como se tornar mais eficaz, ajustando e adaptando o comportamento da equipe.

Com base nesses princípios, no SCRUM, os requisitos do projeto são organizados em uma lista de tarefas, denominada de *Product Backlog*, onde as funcionalidades são ordenadas por prioridades. O conteúdo desta lista é definido

pelo *Product Owner*, pessoa responsável por representar a visão do negócio no projeto.

Nessa metodologia, os projetos são divididos em ciclos com períodos pré-determinados chamados de *Sprints*. O *Sprint* representa um *Time Box* dentro do qual um conjunto de atividades deve ser executado. *O Time Box* é uma técnica para limitar o tempo gasto em uma reunião ou tarefa.

Um *Sprint* tem duração de duas a quatro semanas e, durante um *Sprint*, ocorrem diariamente reuniões curtas, com duração em média de 15 minutos para manter a dinâmica do SCRUM e não atrapalhar a produtividade da equipe (PACHECO, 2010). As reuniões diárias têm como objetivo principal monitorar o andamento do *Sprint*, o estado do processo de cada tarefada realizada pelos membros da equipe, as dificuldades enfrentadas ou problemas identificados, caso haja algum. Com base nas informações coletadas, é atualizado o *taskboard* e o *burndown chat.* O *taskboard* é um quadro dividido em colunas que mostra todo o trabalho que a equipe está desenvolvendo durante o *Sprint*. O *burndown chart* é uma representação gráfica do trabalho restante no Sprint, calculado diariamente.

O *Sprint* possui um objetivo claro e definido, conhecido de toda a equipe. Essas reuniões ocorrem sempre com a presença do *Scrum Master*, sendo que não é o líder da equipe, pois elas são auto-organizadas, mas deve atuar como um mediador dentro da equipe, assegurando que sejam utilizadas corretamente as práticas do SCRUM, além de manter o foco da meta do *Sprint*.

2.3 COMPARAÇÃO ENTRE METODOLOGIAS

Segundo FILHO et al. (2005), os modelos de desenvolvimento tradicionais tentam prever as mudanças, através de técnicas de coletar de requisitos, compreendendo o domínio do problema, e realizando o planejamento, para que as mudanças possam ser controladas durante todo o processo de desenvolvimento do software, enquanto as metodologias de desenvolvimento ágeis tratam as mudanças que possam afetar o processo de desenvolvimento em um curto período de tempo, muitas vezes decorrente a mudança de mercado ou requisitos. Portando a diferença está na maneira de como as mudanças são tratadas durante o projeto.

De acordo com FOWLER (2003), as metodologias ágeis se diferem das metodologias tradicionais em dois aspectos importantes:

- **Metodologias ágeis são adaptativas ao invés de predeterminantes** - As metodologias tradicionais tendem a planejar uma grande parte do processo de desenvolvimento detalhadamente por um longo período de tempo.
- **Métodos ágeis são orientados a pessoas ao invés de serem orientados a processos** - As metodologias tradicionais possuem o objetivo de definir como um processo irá funcionar bem, independente das pessoas envolvidas no processo. Entretanto, os métodos ágeis afirma o papel do processo é dar suporte à equipe de desenvolvimento e seu trabalho, enfatizando os aspectos humanos do desenvolvimento de software.

De acordo com FOWLER (2003), outro ponto importante em relação às duas metodologias, é o processo de documentação, sendo evidente que as metodologias ágeis são menos centradas em documentação, enfatizando uma quantidade menor de documentos para uma determinada atividade, ou seja, são mais voltadas ao código fonte do software.

Segundo NARDI (2010), as metodologias ágeis como o SCRUM é indicado para projetos dinâmicos e suscetíveis a mudanças de requisitos, além disso, promove uma ampla comunicação entre as pessoas envolvidas no desenvolvimento do projeto. Entretanto as metodologias tradicionais, como o PMBOK, são mais adequadas para projetos que necessitam de um forte planejamento e controle nos processos, caracterizado como um modelo de decisão centralizado representado pelo o gerente de projetos. O gerente de projeto é a pessoa designada para atingir os objetivos do projeto (PMBOK, 2008, p. 18). No SCRUM, o papel do gerente de projetos é realizado pelo *Scrum Master,* agindo como um facilitador e não tomando sozinho as decisões, pois a participação da equipe nas decisões é essencial para essa metodologia.

3. SOFTWARE LIVRE

Neste capitulo é apresentado uma revisão bibliográfica sobre os principais conceitos que abrangem o Software Livre e as principais características no seu processo de desenvolvimento.

3.1 CONCEITOS

> Se você tem uma maçã e eu tenho uma maçã, e nós trocamos as maçãs, então você e eu ainda teremos uma maçã. Mas se você tem uma idéia e eu tenho uma idéia, e nós trocamos essas idéias, então cada um de nós terá duas idéias (George Bernard Shaw por AGUIAR et al., 2009 p.4)

Com o avanço da rede mundial de computadores e a expansão da fronteira do **mundo virtual**, que surge o software livre, impulsionado pela capacidade colaborativa da própria Web[2]. Segundo CASTELLS (2003), o Software Livre é considerado por muitos, um movimento tecnológico em um ambiente colaborativo, pois é seu principal meio de comunicação e interação entre os indivíduos que o produz, sendo esse ambiente essencial para o seu desenvolvimento.

De acordo com STALLMAN (2001), podemos definir de maneira simplista um software livre, como uma questão de liberdade, e não de preço. Isso significa que qualquer pessoa tem a liberdade de estudar, mudar e redistribuir o software que utilizar. Um software livre é definido pela sua forma de licenciamento, ou seja, o modo como é regulamentado juridicamente (AGUIAR et al., 2009, p. 79).

Como complemento ao termo software livre, existe o código aberto, que determina o acesso ao código fonte do programa e a liberdade de sua modificação e reuso. “Por reuso de software compreende-se basicamente utilizar artefatos de software existentes no desenvolvimento de um novo software” (KRUEGER, 1992 apud CARDENAS, 2010). O código fonte é o conjunto de palavras escritas de forma ordenada em uma sequencia lógica, contendo instruções em linguagens de

[2] (sigla de *World Wide Web*) *Inform* Teia de alcance mundial, que interliga documentos através de vínculos de hipertexto.

programação (GASPAR, 2008, p. 10).

O termo código aberto foi criado pela OSI (*Open Source Initiative*) e refere-se a software também conhecido por software livre. Segundo Raymond (1999, apud Evangelista, 2006), as duas licenças são concomitantes, enquanto a FSF (*Free Software Foundation*) usa o termo **Software Livre** envolta de um discurso baseado em questões éticas, direitos e liberdade, a OSI usa o termo **Código Aberto** sob um ponto de vista puramente técnico, a fim de evitar questões éticas.

Segundo FALCÃO et al. (2005), compreende-se como software livre aquele que permite aos seus usuários os quatros princípios definido pela FSF de direito ou liberdade, sendo os dois últimos princípios constituem a cláusula de compartilhamento obrigatório, que assume a natureza jurídica de estipulação em favor de terceiros.

1. A liberdade de executar o software, para qualquer uso.
2. A liberdade de estudar o funcionamento de um programa e de adaptá-lo às suas necessidades.
3. A liberdade de redistribuir cópias.
4. A liberdade de aperfeiçoar o programa e de tornar as modificações públicas.

De acordo com a OSI (2011), o termo de distribuição[3] de software de código aberto deve cumprir os seguintes critérios:

1. **Distribuição livre** - A licença não deve restringir de nenhuma maneira a venda ou distribuição do programa gratuitamente, como componente de outro programa ou não.
2. **Código fonte** - O programa deve incluir seu código fonte e deve permitir a sua distribuição também na forma compilada. Se o programa não for distribuído com seu código fonte, deve haver algum meio de se obter o mesmo seja via rede ou com custo apenas de reprodução. O código deve ser legível e inteligível por qualquer programador.
3. **Trabalhos Derivados** - A licença deve permitir modificações e trabalhos derivados, e deve permitir que eles sejam distribuídos sobre os mesmos termos da licença original.

[3] *The Open Source Definition,* Versão 1.9 publicada em 19 de out. 2001.

4. **Integridade do autor do código fonte** - A licença pode restringir o código fonte de ser distribuído em uma forma modificada apenas se a licença permitir a distribuição de arquivos *patch* (de atualização) com o código fonte para o propósito de modificar o programa no momento de sua construção. A licença deve explicitamente permitir a distribuição do programa construído a partir do código fonte modificado. Contudo, a licença pode ainda requerer que programas derivados tenham um nome ou número de versão diferentes do programa original.
5. **Não discriminação contra pessoas ou grupos** - A licença não pode ser discriminatória contra qualquer pessoa ou grupo de pessoas.
6. **Não discriminação contra áreas de atuação** - A licença não deve restringir qualquer pessoa de usar o programa em um ramo específico de atuação. Por exemplo, ela não deve proibir que o programa seja usado em uma empresa, ou de ser usado para pesquisa genética.
7. **Distribuição da Licença** - Os direitos associados ao programa devem ser aplicáveis para todos aqueles cujo programa é redistribuído, sem a necessidade da execução de uma licença adicional para estas partes.
8. **Licença não específica a um produto** - Os direitos associados ao programa não devem depender que o programa seja parte de uma distribuição específica de programas. Se o programa é extraído desta distribuição e usado ou distribuído dentro dos termos da licença do programa, todas as partes para quem o programa é redistribuído devem ter os mesmos direitos que aqueles que são garantidos em conjunção com a distribuição de programas original.
9. **Licença não restrinja outros programas** - A licença não pode colocar restrições em outros programas que são distribuídos juntos com o programa licenciado. Isto é, a licença não pode especificar que todos os programas distribuídos na mesma mídia de armazenamento sejam programas de código aberto.
10. **Licença neutra em relação a tecnologia** - Nenhuma cláusula da licença pode estabelecer uma tecnologia individual, estilo ou interface a ser aplicada no programa

3.2 A COMUNIDADE

> sf (lat communitate) 1 Qualidade daquilo que é comum; comunhão. 2 Participação em comum; sociedade. 3 Social Agremiação de indivíduos que vivem em comum ou têm os mesmos interesses e ideais políticos, religiosos etc. 4 Lugar onde residem esses indivíduos. 5 Comuna. 6 Totalidade dos cidadãos de um país, o Estado. (DICIONÁRIO MICHAELIS, 2009, p. 203)

O desenvolvimento de software livre, em sua maioria, ocorre nas comunidades e alguns projetos têm empresas como suas mantenedoras, com maior ou menor envolvimento da comunidade (BROD e KÄFER, 2009). Em um grupo pode ocorrer a complementação de capacidades, de conhecimentos e de esforços individuais, e a interação entre pessoas com entendimentos, pontos de vista e habilidades complementares (FUKS et al., 2002).

Segundo STEFANUTO et al.(2005), a crescente autonomia das comunidades de software livre levou ao surgimento de inúmeras licenças baseados em suas visões, podendo algumas dessas licenças se distanciarem entre as comunidades do mundo livre, inclusive com a criação de licenças que não outorgam nenhum dos princípios do software livre (liberdade de utilização, cópia, modificação e redistribuição). Entretanto, STEFANUTO et al. (2005) afirmam que essa situação não deverá ocorrer se o software tiver sido originalmente apresentado com uma licença originalmente livre (como a *General Public license,* GPL).

Como base nesse contexto que surge o termo FLOSS (*Free/Libre/Open Source/Software*). O termo foi criado em 2001 por Rishab Aiyer Ghosh e em 2002, foi iniciado um estudo sobre o assunto pela EC (*European Commission*).

> A sigla FLOSS é comumente utilizada internacionalmente como sinônimo de software livre. Como o termo **Livre** (*Free*) em inglês é normalmente utilizado para se referir a uma coisa **gratuita** ou **grátis**, e não à **liberdade** propriamente dita, o termo FLOSS acaba então sendo utilizado para reforçar a dimensão das quatro liberdades garantidas nas licenças de uso que caracterizam um software como **livre** (AGUIAR et al., 2009 p. 4)

Algumas licenças estão disponíveis no site oficial da *Free Software Fundation*, que realiza avaliações dessas licenças, dividindo-as em dois grupos: Licenças compatíveis com GPL ou não compatíveis com a GPL (LINDBERG, 2008 p. 392).

Segundo AGUIAR et al. (2009), o coração do FLOSS é a **comunidade**,

os indivíduos que a compõem e participam em comum consenso nos projetos relacionados à filosofia do software a ser desenvolvido. De acordo PAOLI e ANDREA (2006), as comunidades são centros de esforço e coordenação, incorporando regras de organização. A comunidade tem como objetivos realizar pesquisas, desenvolver tecnologias e promover a inclusão digital utilizando softwares livres. Assim, qualquer individuo que compartilhe as mesmas idéias pode colaborar de diversas formas e o tempo que melhor convir para alcançar os objetivos de um projeto de FLOSS.

De acordo com o PAOLI e ANDREA (2006), os estudos recentes da sociologia do FLOSS demonstram outros aspectos relevantes das comunidades. Esses aspectos ocultos vão a contraponto dos já conhecidos, cujas pessoas participam pelos mesmos interesses ou valores, motivações e abordagens. Os estudos atuais parecem explicar apenas alguns elementos, por falta de uma compreensão da dinâmica do Estilo Bazar (TUOMI 2001, 2004, LIN 2005, apud PAOLI e ANDREA, 2006).

A metáfora criada por RAYMOND (1999), citada frequentemente no estudo de PAOLI e ANDREA (2006), demonstra as divergências entre o processo de desenvolvimento de um software livre e de um software proprietário. Onde, a Catedral corresponde a um processo de desenvolvimento unilateral ou tradicional, comparando a um pastor que entrega sua mensagem aos fiéis sem ser contestado. Já o Bazar, corresponde a um ambiente de troca de ideias (BROD e KÄFER, 2009, p. 8).

De modo geral, independentemente da forma de organização de cada comunidade, todas possuem formas de comunicação, coordenação em menor ou maior grau, discussão, distribuição de tarefas e mecanismo de garantia de qualidade (BROD e KÄFER, 2009 p. 13). A comunicação entre os participantes durante o processo de desenvolvimento de um software livre ocorre de maneira informal e informativa, para divulgação do conhecimento, normalmente por meio eletrônica (PINTO, 2009, p. 40).

Segundo SOFTEX (2005, apud PINTO, 2009), durante o desenvolvimento do software livre, diversas ferramentas são utilizadas de acordo com a atividade a serem realizadas, as ferramentas para controle de versão (CVS - *Concurrent Versions Systems*) e controle de *bugs* são dois exemplos de ferramentas para desenvolvedores.

> O CVS, é um dos sistemas de controle de versão mais antigos, lançado em 1984 por Dick Grune com a intenção de auxiliar o desenvolvimento do ACK (Amsterdam Compiler Kit), desenvolvido em conjunto com seus alunos Erik Baalbergen e Maarten Waage (PISKE et al., 2005 p.6)

Segundo BROD e KÄFER (2009), as ferramentas de controle de versões exercem um papel fundamental dentro das comunidades em Software Livre. Pois é um dos principais fatores para melhora de qualidade e produtividade, além de permite um controle do desenvolvimento distribuído, bem como exposição do código a um imenso número de pessoas. É importante ressaltar que no desenvolvimento de um software livre, poucas ferramentas são utilizadas e grande uniformidade entre os projetos de quais são aplicadas e as ferramentas para controle de versões (CVS) possuem popularidade (REIS, 2003 p. 48).

De acordo com PINTO (2009), muitas vezes, comunidade grandes promovem encontros presenciais, entretanto a comunicação através da internet é o principal meio de interação entre os usuários, as ferramentas utilizadas tanto por desenvolvedores quanto por usuários são listadas em seguida:

a) **Wiki** - Utilizados para identificar um tipo específico de coleção de documentos em hipertexto ou o software colaborativo usado para criá-lo;
b) **Sistemas de controle de *bugs*** - Sistemas de Acompanhamento de defeitos permitem que desenvolvedores ou grupos de desenvolvedores mantenham controle dos bugs pendentes nos seus produtos de forma eficiente;
c) **Listas de E-mail** - Os *mailings* servem para uso de avisos e comunicação em grupos. Podendo também ser usado para suporte online disseminando as respostas das questões levantadas a fim de disseminar o conhecimento com maior eficiência;
d) **Fóruns de Discussão** - são sites ou ferramentas para páginas de Internet destinada a promover debates através de mensagens publicadas abordando uma mesma questão ou assunto especifico.

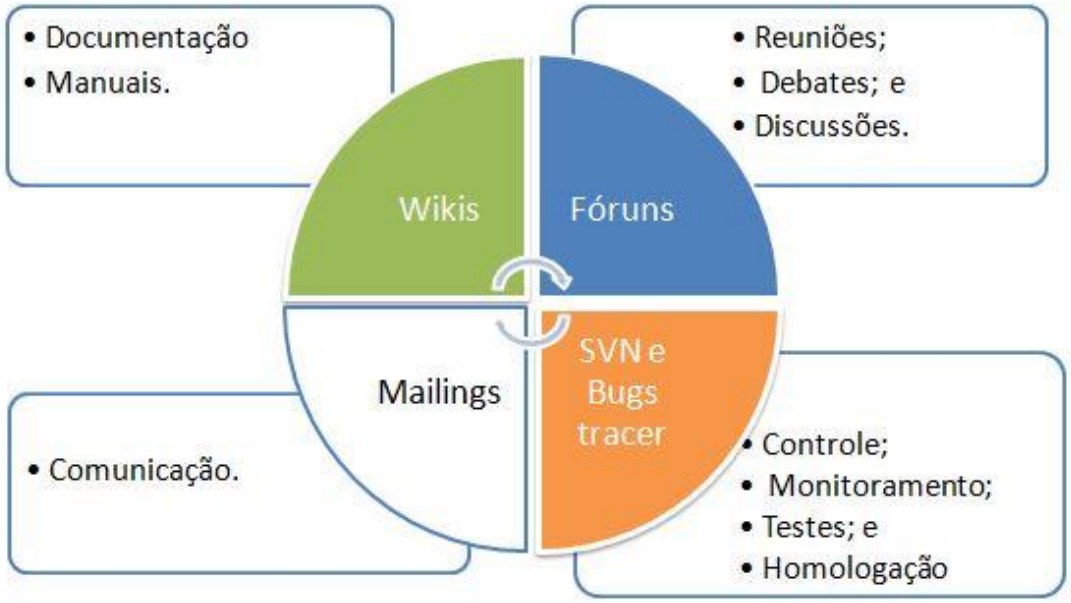

Figura 3. Ferramentas utilizadas durante o desenvolvimento de um software livre.
Fonte: Elaboração dos autores, 2011.

3.3 DESENVOLVIMENTO DE SOFTWARE LIVRE

Segundo SOMMERVILLE (1995, apud PINTO, 2009), o processo de desenvolvimento de um software pode ser definido, como um conjunto de atividades e resultados relacionados que produzem um produto de software. O processo de desenvolvimento de um software tradicional é composto normalmente por sete fases (VIXIE, 1999 apud PINTO, 2009, p. 29):

1. **Levantamento de Requisitos de Mercado** – Os processos de fase estão relacionados com a coleta de requisitos, criação de um documento com necessidades do cliente, e características do produto.
2. **Projeto em Nível de Sistema** – É a discrição formal do software com seus módulos e as interações entre eles.
3. **Projeto Detalhado** – É o documento que descreve a comunicação entre os módulos do software, a *interface* e as dependências entre os módulos.
4. **Implementação** – É o processo de desenvolvimento, codificação do sistema.
5. **Integração** - É a combinação de partes do sistema, podendo ser realizada de forma incremental em paralelo à implementação.

6. **Teste de Campo do Sistema** – É o processo de verificação e correção de falhas durante o uso do sistema dentro da organização.
7. **Suporte** – É o apoio realizado para correção do sistema durante o Teste de Campo, ou após o software ter sido entregue ou distribuído.

O desenvolvimento de Software Livre difere em alguns aspectos do processo de desenvolvimento do software proprietário, não se restringindo somente ao fator de ser desenvolvido em comunidade ou por voluntários. É importante ressaltar que o processo de desenvolvimento de software proprietário é mais rígido e existe uma maior preocupação com a qualidade, além da satisfação dos requisitos solicitados por um cliente (RAYMOND, 1999 apud MARJORIE e PERREIRA, 2006). Além disso, em alguns projetos de software livre não existem especificações de requisitos iniciais, isto se deve ao fato de que grande parte dos projetos surge de motivações pessoais e replicam funcionalidades de algum produto existente (BROD e KÄFER, 2009 p. 9). De acordo com GASSER et al. (2003 apud CARDENAS, 2010, p. 20), as comunidades de software livre geralmente não realizam a coleta de requisitos como sugere o modelo tradicional de desenvolvimento de um software, mas de forma contínua, coletiva e colaborativa.

Segundo REIS (2003), o processo de codificação de um software livre inicia de forma rápida, muitas vezes logo após sua idealização, com grande impulso para lançamento de protótipos funcionais em um curto período de tempo. De acordo com RAYMOND (1999 apud REIS, 2003), o processo de codificação é visto como uma atividade individual e ocorre da seguinte maneira:

1. É criada por uma pessoa ou grupo uma versão inicial do código, testando e desenvolvendo;
2. É realizada a publicação do código inicial através da comunidade ou meio eletrônico pela internet;
3. Outras pessoas se interessam pelo projeto, analisam o código divulgado e caso tenham dúvidas ou sugestões, contatam os desenvolvedores por meio eletrônico;
4. As alterações ao código são enviadas por meio eletrônico aos autores. Essas alterações geralmente são analisadas e discutidas, caso seja

julgada uma alteração positiva, é integrada ao repositório de código e será incluída em uma nova versão do software;

5. Algumas alterações realizadas por membros do círculo de liderança do projeto podem ser integradas ao código fonte sem discussão. No entanto, existe o costume de ser utilizar uma lista de discussão para publicação das alterações realizadas no repositório.

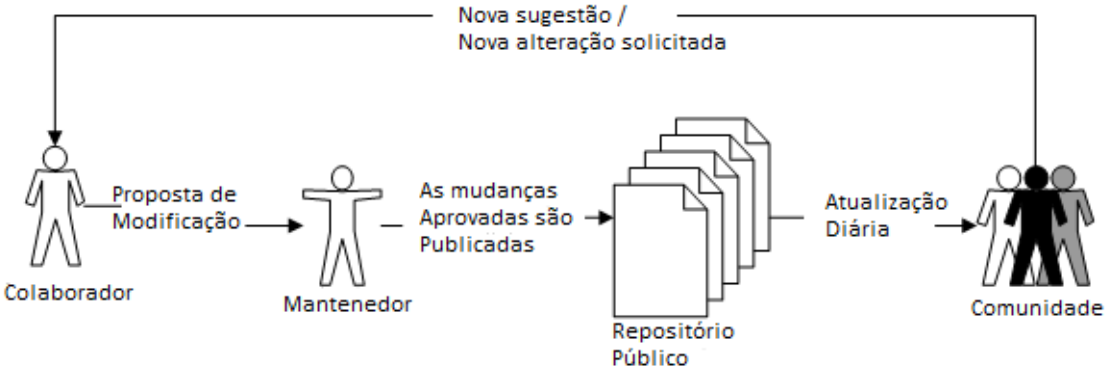

Figura 4. Fluxo de desenvolvimento em um software livre.
Fonte: Adaptado de EARLY, 2010.

Apesar da alta confiabilidade e evolução rápida dos projetos dentro de uma comunidade ativa, os projetos executados por ela apresentam um processo desenvolvimento aparentemente caótico, com características particulares entre esses projetos (REIS, 2003, p. 4):

a) Descentralização dos processos e organização;
b) Política liberal em relação a alterações;
c) Discussão aberta sobre todo assunto técnico pertinente ao projeto;
d) Maior parte dos projetos utilizam um mínimo de infraestrutura, provida pela Internet;
e) O email é a ferramenta mais utilizada para comunicação entre os indivíduos da comunidade.

De acordo com MARJORIE e PERREIRA (2006 p. 8), os requisitos funcionais do sistema são alterados de forma dinâmica, dado que os papéis de participante e de usuário se misturam. Dessa forma, muitos projetos não possuem um único líder, variando entre os usuários em cada etapa do desenvolvimento, com maior ou menor grau de responsabilidade. Segundo TAURION (2004 apud

MARJORIE e PERREIRA, 2006), apesar da descentralização da coordenação no desenvolvimento do software livre, a estruturação e organização do projeto é realizado por um líder ou mantenedor do projeto. O líder juntamente com outros colaboradores, é o responsável pela coordenação do projeto, seleção das melhores sugestões e códigos de voluntários.

Segundo ARAKAKI (2008), de forma similar a gerência de projetos, nas comunidades é possível identificar papéis com suas contribuições, permissões e acessos as ferramentas de suporte:

1. Usuários não-ativos - São pessoas que apenas fazem uso do software, sem contribuição direta para a comunidade;
2. Usuários ativos - São pessoas que possuem uma participação maior nas comunidades, participando das discussões;
3. Relatores de *bugs* - Participam do desenvolvimento, contribuindo com a localização de problemas e a especificações destes.
4. Corretores de *bugs* - Corrigem os problemas relatados e enviar trechos do código alterado para avaliação;
5. Desenvolvedores esporádicos - São pessoas com conhecimento em desenvolvimento, que contribuem de maneira ocasional para o projeto, não estando totalmente comprometidos com a comunidade;
6. Testadores – Realizam testes de software, visando a melhoria na qualidade do software;
7. Desenvolvedores ativos – São desenvolvedores que contribuem de maneira constante para o projeto, tendo responsabilidade por grande parte do código fonte desenvolvida;
8. Documentadores – Elaboram documentos e manuais para o projeto, sendo relacionados tanto ao projeto quanto à comunidade;
9. Tradutores - Realizam a tradução do software, documentos, páginas do projeto em outros idiomas, possibilitando sua internacionalização;
10. Membros de confiança – Na maioria das comunidades, os membros de confiança ou do núcleo são relativamente pequenos. Essas pessoas fornecem maior parte das contribuições ao projeto, participando desde o inicio do seu desenvolvimento;
11. Líderes de módulos - São pessoas que visam reduzir a carga da responsabilidade dos membros de confiança. Sendo responsáveis por

módulos do software desenvolvido por uma comunidade;

12. Conselheiros/Patrocinadores - Alguns projetos de software livre possuem patrocinadores externos e podem participar da comunidade como conselheiros, participando ativamente do processo de decisão;
13. Líderes de projetos – Os líderes geralmente são pessoas que iniciaram o projeto, porém sua liderança pode ser repassada para outra pessoa;
14. Membros do comitê Administrativo – Este papel existe somente nas maiores comunidades que adotam um comitê para executar liderança da mesma. Basicamente definem políticas internas, padrões para tomada de decisões na comunidade (tal como, a decisão baseada em votos).

É importante ressaltar que a evolução do “status” dos participantes em uma comunidade de software livre é definido na sua maioria, por um processo de meritocracia (ARAKAKI, 2009, p. 62). Neste sentido, a produção de trabalhos relevantes na comunidade implica em ganho de respeito, status e influência. Diferente do modelo tradicional, os papéis de liderança e gerenciamento podem variar no decorrer do desenvolvimento ou amadurecimento da comunidade.

3.4 RELAÇÃO ENTRE GERÊNCIA DE PROJETOS E SOFTWARE LIVRE

Segundo ARAKAKI (2008), muitas comunidades de software livre adotam alguma forma de gerenciamento de projeto, apesar da estrutura organizacional informal na qual são constituídas as comunidades. O gerenciamento de projeto se difere do gerenciamento do desenvolvimento de um software proprietário, diante as características do software livre. ARAKAKI (2008), afirma que é possível identificar, nos projetos de software, padrões de estrutura de controle bem definidos para gerenciamento. RUDZI e JONSON (2003, apud PINTO, 2009) concluíram, em sua pesquisa, que as práticas de gerenciamento utilizadas nas comunidades visam a conversão do conhecimento, para suporte à criação e ao compartilhamento do conhecimento. Dessa forma é possível identificar práticas da gestão do conhecimento nessas comunidades.

De acordo com ARAKAKI (2008), é possível identificar nas comunidades

de software livre padrões de gerenciamento de projetos, categorizados em oitos tipos de gerências:

1. Gerência de Requisitos - Essa gerência é caracterizada pela definição e compreensão das funcionas que o software deve cumprir. Esta etapa não segue rigorosamente os métodos tradicionais de engenharia de requisitos, pois se baseia na troca de ideias entre os colaboradores;
2. Gerência de Lançamento de Versões do Software - Consiste na disponibilização de versões, estáveis ou não, do software desenvolvimento pela comunidade em formatos de distribuição que facilitam o acesso;
3. Gerência de Evolução Orientada a *Bugs* - É a prática de amadurecimento do projeto em decorrência das falhas e erros relatados, utilizando sistema de rastreamento e acompanhamento de mudanças;
4. Gerência de Qualidade - Esse processo assegura que o software desenvolvido cumpre as especificações e atenda às necessidades, garantindo que o software está sendo construído corretamente;
5. Gerência de código-fonte - Essa gerência define como registrar e processar as alterações propostas para o software, além de, como relacioná-las aos seus componentes para identificar diferentes versões do código;
6. Gerência de Coordenação da Comunidade - É o gerenciamento de recursos em uma comunidade, solucionando problemas técnicos e não-técnicos utilizando os recursos de maneira eficaz.
7. Gerência de Comunicação - Definição de métodos e ferramentas que viabilize a interação entre os participantes da comunidade;
8. Gerência de Documentação - As documentações disponibilizada nas comunidades se distância em modelos com alguma formalização, entretanto, os documentos gerado durante o processo de desenvolvimento apresenta-se como documentos relevantes à a comunidade (desenvolvimento, meios de integração e processos). Além da documentação relevante aos usuários, que trazem informações sobre aspectos técnicos e funcionais.

Segundo ARAKAKI (2008), é possível correlacionar as áreas de

conhecimento apontadas pelo PMBOK com as práticas de gerência identificadas nas comunidades de software livre, conforme FIGURA 5.

	Gerências do PMBOK								
	Integração	Escopo	Tempo	Custo	Qualidade	Recursos Humanos	Comunicação	Riscos	Aquisição
Requisitos	X	X			X		X	X	
Lançamento de Versões			X		X		X		
Evolução Orientada a Bugs					X		X		
Qualidade					X		X		
Código-Fonte		X			X	X	X		
Coordenação	X	X	X	X	X	X	X	X	X
Comunicação	X	X			X	X	X		
Documentação					X		X		

Figura 5. Correlação das Gerências PMBOK e as de Projetos de Software Livre.
Fonte: ARAKAKI, 2008, p. 91.

De acordo com ARAKAKI (2008), a gerência de coordenação das comunidades tem uma forte relação com todas as áreas de conhecimento do PMBOK. Apesar dos projetos de software livre não apresentarem uma estrutura formal, as suas gerências são voltadas a comunicação e qualidade. Isso implica que todas as gerências do software livre têm ligação com a gerência de qualidade e comunicação do PMBOK. Entretanto, a gerência de custo e aquisição não possui relações significativas. Dessa forma, não são todas as práticas da gerência de projetos que são adotadas pelas comunidades de software livre, devido às diferenças entre o seu processo de desenvolvimento e o modelo tradicional de desenvolvimento de software.

4. FERRAMENTAS DE GERÊNCIA DE PROJETOS

Esta parte do trabalho é composta pela seleção de ferramentas de gerência de projetos para análise, definição dos critérios avaliativos e comparação de resultados.

4.1 CONCEITOS

Segundo MARJORIE e PERREIRA (2006), a adoção de ferramentas durante o ciclo de desenvolvimento de um software, pode conduzir a um sistema com menor número de erros e maior grau de qualidade. A aplicação correta de ferramentas, alinhada ao conhecimento, pode aumentar as chances de um projeto alcançar seu objetivo (CAVALCANTI e CASTRO, 2007, p. 13).

De acordo com ARAUJO e QUINTÃO (2010), conhecer ferramentas de gerenciamento de projetos é cada vez mais importante na medida em que uma organização avança em seu nível de maturidade. Segundo CAMPOS e LIMA (2008), as ferramentas para gerência de projetos utilizam-se de modelos e padrões existentes da engenharia de software, que visam a criação dos planos de gerenciamento.

Segundo o PMBOK (2004), as ferramentas para gerência de projetos caracterizam um tipo de software especificamente desenvolvido para auxiliar no planejamento, monitoramento e controle de projetos. De acordo com FREITAS e MOURA (2004), a maioria das ferramentas para gerência de projetos apresenta similaridade em suas funcionalidades, pois foram desenvolvidas para atender os projetos de propósito em geral. Entretanto, algumas focam em áreas específicas como gerência de custos, gerência de atividades, gerência de comunicação e integração da equipe, dentre outras.

4.2 FORMALIZAÇÃO DOS CRITÉRIOS AVALIATIVOS

O objetivo da definição desses critérios é auxiliar na seleção das ferramentas, baseado nas características identificadas através da revisão bibliográfica realizada nos capítulos de Gerência de Projeto e Software Livre. Além disso, viabilizarão a análise e comparação das ferramentas selecionadas.

Nesse contexto, os seguintes critérios foram definidos a partir do conhecimento acumulado do estudo das práticas recomendadas pelo PMBOK e das práticas de gerência de projetos em comunidades de desenvolvimento de software livre; Acredita-se que esses critérios permitirão selecionar e comparar algumas ferramentas livres de gerência de projetos à luz das características e práticas dessas comunidades.

- **Critério 1: A ferramenta deve ser um Software Livre** - As ferramentas analisadas devem ter seu código-fonte aberto disponível para análise e para possíveis propostas de modificações de acordo com os princípios do software livre, possibilitando assim, o seu reuso, segundo FALCÃO et al.(2005).
- **Critério 2: A ferramenta deve apresentar uma interface Web** - A justificativa para esse critério baseia-se no modelo de desenvolvimento de um software livre, em sua maioria ocorre em comunidades distribuídas geograficamente, tendo a web como o principal meio de comunicação, interação e publicação (BROD e KÄFER, 2009).
- **Critério 3: A ferramenta deve possibilitar a gerência de projetos** – Como já citado, segundo ARAKAKI (2008), é possível identificar nos projetos de software livre, padrões de gerenciamento e estrutura de controle, possibilitando o correlacionamento das práticas de gerência realizadas nas comunidades de software livre com as nove gerências do PMBOK. Segundo o PMI, as gerências caracterizam os principais aspectos envolvidos em um projeto e no seu gerenciamento. Assim as ferramentas selecionadas, devem conter recursos que possibilitem:
    1. Gerenciamento de Integração;
    2. Gerenciamento de Escopo;
    3. Gerenciamento do Tempo;

4. Gerenciamento de Custo;
5. Gerenciamento da Qualidade;
6. Gerenciamento de Recursos Humanos;
7. Gerenciamento de Comunicações;
8. Gerenciamento de Risco;
9. Gerenciamento de Aquisições.

- **Critério 4: A ferramenta deve possibilitar que uma atividade seja realizada por uma ou mais pessoas de forma colaborativa** - Segundo NUNES e STAA (2007), o desenvolvimento de um software livre na maioria das vezes é realizado pelo interesse no compartilhamento do conhecimento, podendo um indivíduo interagir em diversas áreas de conhecimento em um projeto, com maior aptidão ou interesse.
- **Critério 5: A ferramenta deve possibilitar a obtenção de requisitos em qualquer fase do projeto** - Segundo MARJORIE e PERREIRA (2006), a maioria dos projetos em software livre não possui uma especificação de requisitos formal (e inicial) nos moldes da engenharia de requisitos tradicional. Os colaboradores passam a ser as principais fontes de obtenção dos requisitos a medida que o software vai sendo desenvolvido, segundo NUNES e STAA (2007, p.8).

4.3 SELEÇÃO DE FERRAMENTAS

Pesquisa exploratória[4] no site *Source Forge*, repositório publico de projetos (em sua maioria de código aberto), retornou cerca de 4298 projetos de gerenciamento de projetos[5] disponíveis no repositório. O *Source Forge* foi selecionado como base de pesquisa por ser um dos principais ambientes de desenvolvimento colaborativo, utilizado por diversas comunidades (ARAKAKI, 2008, p. 19).

Os resultados da pesquisa foram classificados por ordem de relevância, conforme classificação realizada pelo próprio *Source Forge* que utiliza os seguintes

[4] Pesquisa realizada em 17 de junho de 2011.
[5] Ferramentas categorizadas como *Project Management* pelo site sourceforge.net.

critérios para essa classificação:

- Informações estatísticas: número de downloads da semana e recomendações pelos usuários;
- Popularidade do projeto: É considerado o número de visitas à página do projeto e contribuições em geral realizadas ao projeto pela comunidade;
- Maturidade: nível de estabilidade do projeto e da comunidade;
- Relevância: Número de referências do projeto e documentos sobre o projeto de uma determinada comunidade.

Considerando os critérios de classificação do *Source Forge* e aplicando os dois primeiros critérios formalizados anteriormente (A ferramenta deve ser um software livre e apresentar uma interface Web), obtivermos os seguintes projetos como resultados da seleção:

1.]project-open[- Project Management
2. Feng Office - Your World Wide Office
3. dotProject
4. EGroupware Enterprise Collaboration
5. TeamLab

4.3.1]project-open[- Project Management

O sistema *project open* é uma aplicação web, com recursos para gerenciamento de projetos e ERP(*Enterprise Resource Planning*). O projeto é mantido por um grupo de desenvolvedores de software, com sede em Espanha e Alemanha, foi inicialmente desenvolvimento por Philip Greenspun e posteriormente assumido por Frank Bergmann e Klaus Hofeditz, que acrescentaram novas extensões e funcionalidades ao sistema. O projeto foi oficialmente publicado em 2003 através do repositório publico *Source Forge.*

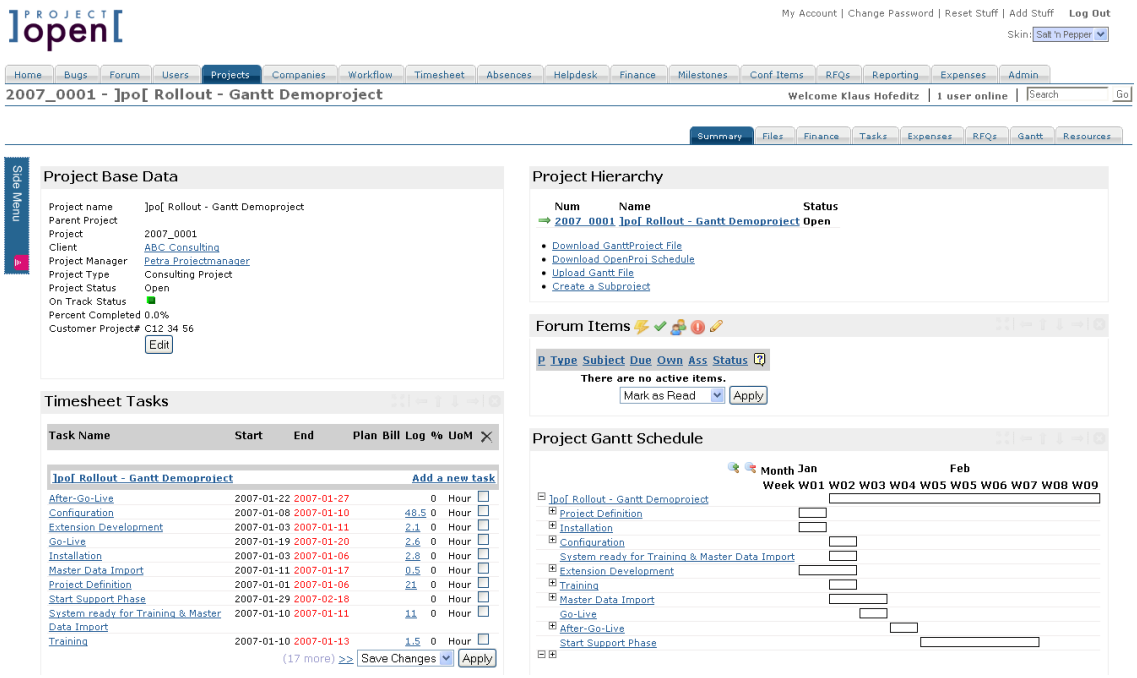

Figura 6. *Screenshot* da ferramenta *]project-open[– Project-Open Development Team*.

4.3.2 Feng Office

O sistema *Feng Office* é uma aplicação escrita em linguagem de programação PHP (Hypertext Preprocessor). Foi inicialmente projetado na Universidad de la República do Uruguai, por Conrado Viña, como trabalho de conclusão de curso. O sistema foi inicialmente desenvolvido por Marcos Saiz e Ignacio de Soto e publicado em 2008. Inicialmente o projeto se chamava OpenGoo, entretanto após a fundação da empresa *Feng Office* por seus criadores, passou a ter o mesmo nome da empresa mantedora do projeto.

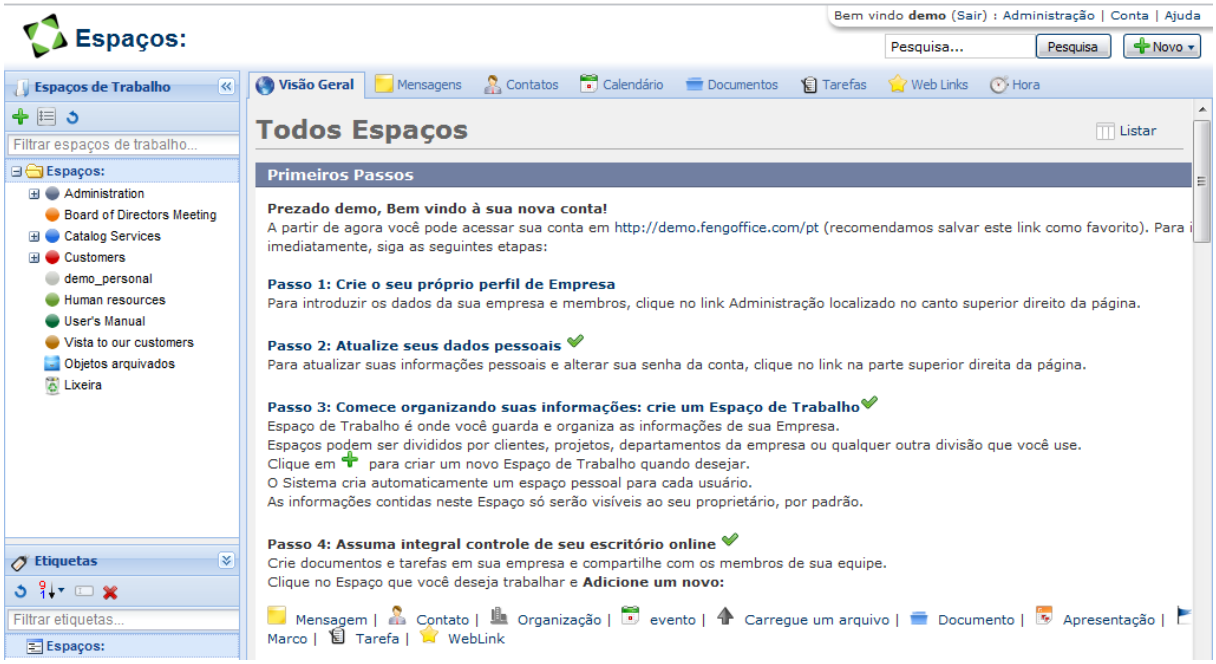

Figura 7. *Screenshot* da ferramenta *Feng Office* versão demo – *Feng Office*.

4.3.3 dotProject

O sistema dotProject é uma aplicação web escrita em linguagem de programação PHP(*Hypertext Preprocessor*). Originalmente iniciado em 2000 por dotmarketing.org, o projeto foi desenvolvido como uma alternativa em software livre a ferramentas comerciais existente. Atualmente é mantida por sua própria comunidade e possui como principais desenvolvedores Adam Donnison, Karen Chisholm, Gregor Erhardt, Ivan Peevski, Eamon Brosnan e Benjamin Young.

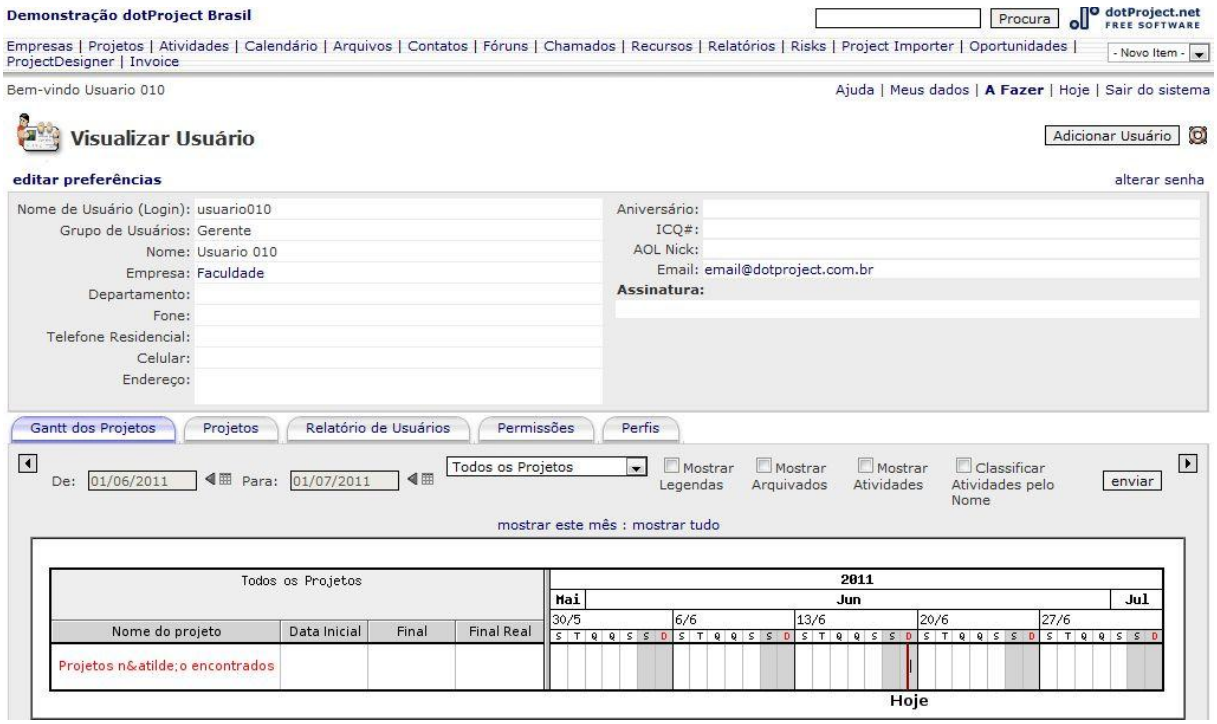

Figura 8. *Screenshot* da ferramenta *dotProject* versão demo – *dotProject*.

4.3.4 EGroupware Enterprise Collaboration

O sistema eGroupware é uma aplicação web escrita em linguagem de programação PHP(*Hypertext Preprocessor*). O projeto iniciou-se em 2000, com desenvolvimento do projeto phpgroupware[6], que foi baseado no webdistro. Em 2003 houve uma separação na comunidade e iniciou-se o desenvolvido do *eGroupware* com intuito de ter um caráter mais comunitário no seu desenvolvimento em comparação aos seus antecessores. Diferente de outros projetos, o *eGroupware* adotou em 2005 uma constituição em sua comunidade, que garante a liberdade e segurança para a sua comunidade estabelecer eleições dos seus administradores. Atualmente possui como principal mantedor do projeto Aleksey Avdeev

[6] O phpGroupware é uma ferramenta de escritório virtual e colaboração escrita em PHP.

Figura 9. *Screenshot* da ferramenta *eGroupware – EGroupware Content Management System*.

4.3.5 TeamLab

O sistema *TeamLab* é uma aplicação web escrita em linguagem de programação ASP.NET, desenvolvida pela empresa *Ascensio System* e licenciado sob a licença GNU GPLv3. Seu projeto foi iniciado em dezembro de 2009 como uma simples plataforma de colaboração online, na qual abrangia recursos de mídia social (blog, fórum, wiki, favoritos). A partir de junho 2010, novos recursos foram sendo adicionados, visando possibilitar o gerenciamento de projetos.

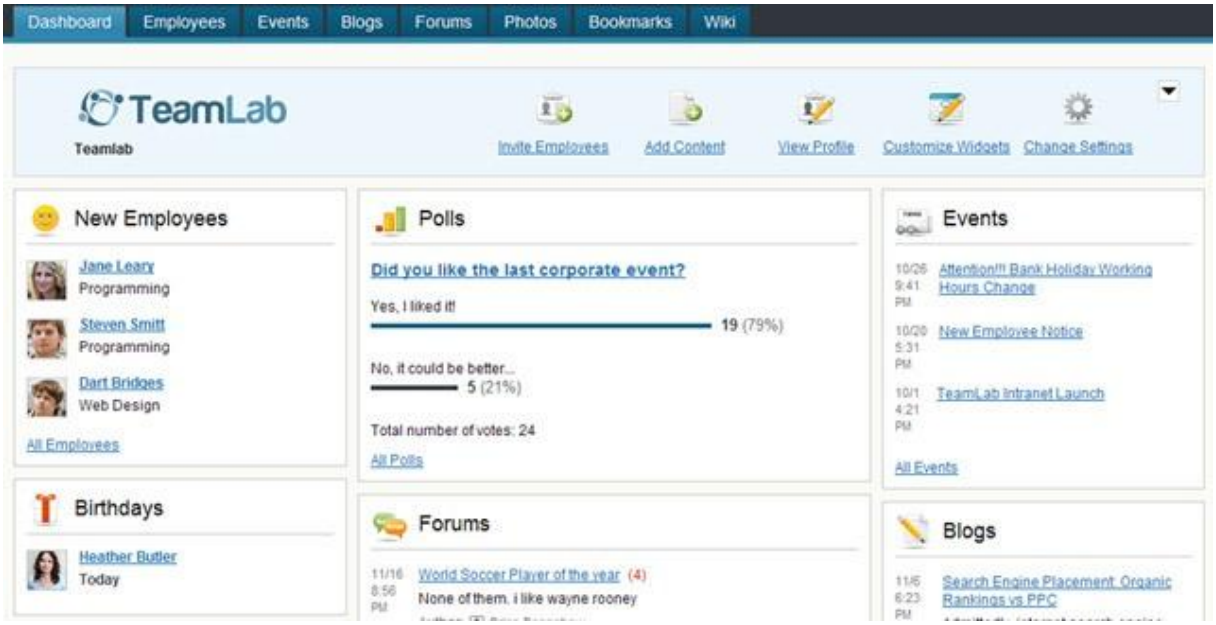

Figura 10. *Screenshot* da ferramenta *TeamLab – Ascensio System*.

4.4 ANÁLISE DE FERRAMENTAS SOB A ÓPTICA DA GERÊNCIA DE PROJETOS

Como apresentado anteriormente, foi realizada a filtragem de acordo com os dois primeiros critérios para seleção dos projetos de ferramentas em gerência de projetos, destinado a análise. Entretanto, devido aos critérios de seleção adotados e o número de projetos de ferramentas existentes no *Source Forge*, consideramos o critério de relevância dos projetos para as comunidades, de acordo com a classificação do próprio *Source Forge*, para escolha dos projetos. A escolha de cinco projetos de ferramentas foi motivada principalmente devido ao grau de relevância dessas ferramentas em comparação às demais, segundo classificação do *Source Forge*. Desta maneira, os cincos projetos de ferramentas foram selecionados. São também apresentadas algumas informações a respeito dos mesmos.

Para identificar os recursos de gerência nos projetos de ferramentas, segundo definição do critério três (A ferramenta deve possibilitar a gerência de projetos), procuraram-se funcionalidades que atendessem ou fornecessem suporte para cada processo de desenvolvimento em sua respectiva área de conhecimento (PMBOK, 2004). Para levantamento das informações, inicialmente considerou-se a existência de trabalhos científicos e acadêmicos. Em um segundo momento, por meio de estudo direto nas próprias ferramentas, realizou-se uma análise para descoberta de recursos de gerência.

É adotada a simbologia descrita na TABELA 3 para melhor compreensão e entendimento da análise a ser realizada das funcionalidades e recursos disponíveis nas ferramentas selecionadas, em relação às gerências de projetos e seus respectivos processos segundo PMBOK (2004). Assim, a simbologia adotada será utilizada na análise de cada grupo gerência e os processos que a compõem. Ao final da análise de cada grupo de gerência é fornecida uma tabela-resumo sobre o resultado referente aos processos da gerência estudada, para oferecer uma visão de cada grupo estudado e uma compreensão dos resultados forma objetiva dos diferentes processos que influenciaram nos resultados da análise realizada em cada ferramenta.

Tabela 3. Simbologia para análise das funcionalidades dos projetos de ferramentas.

Símbolo	Descrição
	Fornece recursos que atendem ao processo de gerência em questão
	Fornece recursos que atendem parcialmente ao processo de gerência em questão
	Não foram identificados recursos que atendam o processo de gerência em questão

A partir dessa etapa, inicia-se a análise dos projetos de ferramentas sob óptica das práticas de gerência de projetos descritas no PMBOK.

4.4.1 Gerenciamento de Integração

No *dotProject* e *]project-open[,* identificamos que ao criar um novo projeto, os parâmetros de cadastro do projeto (Empresa, Responsável pelo Projeto, Data de inicio e fim prevista para o projeto, Prioridade, Situação e Descrição) simulam um termo de abertura de projeto. A criação de um novo projeto no *]project-open[* requer o preenchimento de um número maior de campos. O Plano de gerenciamento do projeto e monitoramento pode ser realizado através do cadastro de tarefas, ambas as ferramentas possibilitam a definição de uma estrutura similar a uma EAP[7] para as atividades a serem realizadas. O controle de mudanças pode ser realizado através dos registros de *logs*, de cada tarefa. O *dotProject* possibilita que o controle de mudanças seja realizado com o software Mantis[8], através de módulo adicional de integração.

Nos projetos *Feng Office* e *TeamLab*, a criação de um projeto e o cadastro de empresas é realizada de forma superficial, sem a necessidade de muitos detalhes, bem como as tarefas. O monitoramento é realizado através de

[7] A Estrutura Analítica do Projeto (EAP) é uma decomposição hierárquica orientada às entregas do trabalho a ser executado pela equipe, sendo que cada nível descendente da EAP representa uma definição gradualmente mais detalhada da definição do trabalho do projeto, onde os componentes de nível mais baixo da EAP são chamados de pacotes de trabalho (Viégas et al., 2010, p. 26).

[8] Mantis Bug Tracker é uma ferramenta baseada na web escrita em linguagem de programação PHP e licenciado sob licença GPL.

listas ou gráficos, além de não ser possível determinar a prioridade entre as tarefas, assim não é possível realizar uma gerência de integração por completo através dessas duas ferramentas.

O *EGroupware* funciona de forma similar ao *dotProject*, referente ao cadastro de tarefas e projetos, entretanto no *EGroupware* é possível realizar também o cadastro de subprojetos. Não é possível a criação de uma estrutura para a lista de tarefas e o controle de mudanças é feito somente através do histórico de alterações.

A TABELA 4 apresenta a relação dos recursos identificados em cada uma das ferramentas e os relaciona aos processos da gerência de integração.

Tabela 4. Resultado da análise da gerência de integração.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Desenvolver o termo de abertura	●	○	●	●	○
Desenvolver declaração de Escopo	◐	○	◐	◐	○
Desenvolver plano de gerenciamento	●	◐	●	●	◐
Gerenciar a execução do projeto	●	◐	●	●	◐
Monitorar e controlar as atividades	●	◐	●	●	◐
Realizar o controle integrado de mudanças	●	◐	●	◐	◐
Encerrar o projeto ou a fase.	◐	○	◐	◐	○

4.4.2 Gerenciamento do Escopo

Esse grupo de gerência é composto por muitos processos que tratam a coleta de requisitos. Essas atividades são realizadas essencialmente pelo gerente de projetos em conjunto com a equipe, tradicionalmente realizadas de forma

independente de ferramentas. Entretanto, algumas ferramentas oferecem recursos (Fóruns, lista de discussões) que visam auxiliar na coleta de requisitos para grupo de trabalhos geograficamente dispersos. As ferramentas *EGroupware* e *TeamLab* por padrão contêm esses recursos adicionais para a coleta e aprendizado, sem a necessidade de módulos adicionais, além de fornecer sistema de votação e debate para os requisitos cadastrados. Na ferramenta *Feng Office* não foram identificados recursos como fórum ou lista de discussões, entretanto é possível importar documentos de registro das entrevistas, dinâmicas de grupo e *brainstorms*, possibilitando seu armazenamento em um repositório. Há ainda um recurso interessante no *Feng Office*: a possibilidade de criação de campos customizados para uso no registro de novos projetos, que podem ser utilizados na definição do escopo do projeto.

Nos processos de criação da EAP, como dito anteriormente, somente as ferramentas *dotProject* e *]Project-open[* possibilitam a criação de uma estrutura analítica, através do cadastro de tarefas e o controle de mudanças. As demais ferramentas possuem somente o controle das alterações através de registros de *tickets* ou *logs*. Assim, o controle de escopo ocorrerá basicamente através do controle de alteração realizado nas tarefas.

Tabela 5. Resultado da análise da gerência de escopo.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Planejamento do escopo	●	○	◐	●	◐
Definição do escopo	◐	○	◐	●	◐
Criar EAP	●	◐	●	◐	◐
Verificação do escopo	●	◐	●	◐	◐
Controle do escopo	●	◐	●	◐	◐

4.4.3 Gerenciamento do Tempo

Todas as ferramentas possuem recursos para o controle de tempo das tarefas a serem realizadas, estimando o tempo de início e de fim de cada tarefa.

Entretanto, existem diferenças de como algumas tratam essa informação. No *dotProject* e *]Project-open[*, as tarefas possuem um sequência segundo sua estrutura analítica e controle de dependência entre as atividades a serem realizadas, além de fornecer um gráfico de G*antt* [9] para auxiliar no controle do cronograma.

O *EGroupware* fornece recursos como gráfico de G*antt* e relatórios customizados, além de possibilitar o controle de custo e tempo em relatório específicos. As demais ferramentas possuem recursos similares, exceto pela geração de gráfico de G*antt*, uma vez que esse recurso é substituído por relatórios de tarefas e histórico de tempo. No *Feng Office,* somente o controle de tempo é realizado, pois não existem campos para o controle de custo, uma vez que a ferramenta possibilita somente o cadastro de custo através de campos customizados. Por fim, não foi identificado recursos para controle de custos no *TeamLab*.

Tabela 6. Resultado da análise da gerência de tempo.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Definição das atividades	●	●	●	●	●
Sequencialmente de atividades	●	◐	●	◐	◐
Estimativa de recursos da atividade	●	◐	●	●	○
Estimativa de duração da atividade	●	●	●	●	●
Desenvolvimento do cronograma	●	○	●	●	○
Controle do cronograma	●	○	●	●	○

4.4.4 Gerenciamento de Custo

Segundo o PMBOK, o gerenciamento de custo do projeto é composto por três processos (Estimar custos, Determinar orçamento e Controlar custos). Já foi

[9] O diagrama de Gantt é um gráfico usado para ilustrar o avanço das diferentes etapas de um projeto.

identificado na etapa anterior que somente as ferramentas *]Project-open[*, *dotProject* e *EGroupware* permitem por padrão a entrada de valores monetários para as tarefas cadastradas e relatórios específicos que possibilitem o acompanhamento. Por essa razão, foi realizada a análise somente destes três projetos.

Assim, verificamos que o *dotProject* possui funcionalidades básicas, que visam atender o controle de finanças, sendo possível a instalação de módulos adicionais (*Reports* , *Invoices*), para auxiliar no orçamento e relatório de custo. Não obstante, o *EGroupware* fornece relatórios específicos para os custos de um projeto, baseado nos valores informado em cada tarefa, de forma similar ao *dotProject*. As duas ferramentas não realizam os somatórios automáticos de custos previstos, e nem permitem a inserção de valores de reserva e outros que possam ser necessários, focando essencialmente no custo de cada tarefa.

O *]Project-open[* em comparação as outras duas ferramentas, contém mais recursos, que visam auxiliar no controle de custos e orçamento. Essa ferramenta contém recursos financeiros avançados, possibilitando os registros de despesas agrupados por fornecedor e cliente, além dos cadastros de despesas adicionais, controle da taxa de câmbio e controle de aprovação para despesas através do centro do custos.

Por fim, é importante ressaltar que existem versões customizadas do dotProject (UFSC/Cyclops[10]), que apresentam funcionalidades avançadas para o controle de finanças, visando uma melhoria no controle de custos, sendo que as versões customizadas não fazem parte desse estudo.

Tabela 7. Resultado da análise da gerência de custo.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Estimativa de custos	●	◐	●	◐	○
Orçamentação	●	○	◐	◐	○
Controle de custos	●	○	◐	◐	○

[10] A versão customizada do dotProject, denominada dotProject+, foi desenvolvida e disponibilizada pelo grupo Cyclops da Universidade Federal do Estado de Santa Catarina com o objetivo de atender dos resultados esperados segundo os modelos de CMMI-DEV (*Capability Maturity Model Integration for Development*) e MPS.BR(Melhoria de Processos do Software Brasileiro).

4.4.5 Gerenciamento da Qualidade

Segundo VIÉGAS (2010), o planejamento da qualidade é a realização do processo de identificação dos requisitos de qualidade do projeto, além da definição de que modo o projeto demonstrará conformidade, sendo essa etapa realizada em paralela com os demais processos do projeto. Além disso, o planejamento da qualidade é um processo que faz uso dos conhecimentos do gerente de projeto e das pessoas envolvidas.

Nenhuma ferramenta selecionada possui recursos específicos para o gerenciamento de qualidade, entretanto todas fornecem informações que podem ser utilizadas para tal monitoramento. O processo de garantia da qualidade será beneficiado a medida que as informações forem atualizadas e acompanhadas diariamente pelo gerente de projeto ou colaboradores. Assim, consideramos que as ferramentas atendem parcialmente os requisitos de qualidade, pois oferecem formas de monitoramento das tarefas e prazos, possibilitando também a criação de um plano de qualidade através de tarefas especificas.

Tabela 8. Resultado da análise da gerência de qualidade.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Planejamento da qualidade	◐	◐	◐	◐	◐
Realizar a garantia da qualidade	◐	◐	◐	◐	◐
Realizar o controle da qualidade	◐	◐	◐	◐	◐

4.4.6 Gerenciamento de Recursos Humanos

O *]Project-open[, dotProject, TeamLab* e *EGroupware* oferecem suporte para o planejamento dos recursos humanos, possibilitando o cadastro de papéis de cada colaborador, necessitando assim que haja uma pessoa responsável pelo cadastrado de forma adequada. Com base no cadastramento dos colaboradores, é possível posteriormente mobilizar os recursos em cada atividade, realizando o

controle de recursos através dos gráficos de G*antt (]Project-open[, dotProject e EGroupware)* e relatórios de participação nas tarefas. Não obstante, o *]Project-open[* contém recursos adicionais para o gerenciamento de recursos humanos, que permitem um maior controle, em comparação às ferramentas *dotProject*, *EGroupware* e *TeamLab*.

Como em outros gerenciamentos, esse depende de um responsável pelo cadastro dos colaboradores e definição da equipe, e as ferramentas fornecem em maior ou menor grau recursos para a gerência de recursos humanos. Na ferramenta *Feng Office*, o cadastro de colaboradores é superficial, e as tarefas não são designadas, mas assumidas pelos colaboradores.

Tabela 9. Resultado da análise da gerência de recursos humanos.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Planejamento de recursos humanos	●	○	●	◐	◐
Contratar ou mobilizar a equipe do projeto	●	○	◐	◐	◐
Desenvolver a equipe do projeto	●	○	●	●	◐
Gerenciar a equipe do projeto	●	○	●	●	○

4.4.7 Gerenciamento das Comunicações

Dentre as ferramentas analisadas, todas possuem recursos que identificam as pessoas envolvidas no projeto, podendo ser vinculada diretamente a um projeto e ser notificadas através de meio eletrônico sobre as atualizações do projeto. É possível também a criação de grupos ou departamento, organizando assim as partes interessadas de forma mais eficiente. Dessa maneira, qualquer pessoa devidamente cadastrada e vinculada a um projeto ou tarefa pode ser mantida atualizada sobre o andamento do projeto.

Entre os processos desse gerenciamento, percebemos que somente o processo de gerenciar as expectativas das partes interessadas depende

essencialmente de uma pessoa responsável, nesse caso o gerente de projeto. Além disso, as ferramentas possibilitam a análise de desempenho em menor ou maior grau, dependendo da ferramenta, nesse processo as ferramentas *Feng Office* e *TeamLab* possuem menos informações a serem exibidas, devido às suas limitações no gerenciamento de recursos humanos.

Tabela 10. Resultado da análise da gerência das comunicações.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Planejamento das comunicações	●	●	●	●	●
Distribuição das informações	●	●	●	●	●
Relatório de desempenho	●	◐	●	●	◐
Gerenciar as partes interessadas	◐	◐	◐	◐	◐

4.4.8 Gerenciamento de Riscos

Esse gerenciamento é composto por seis processos (Planejamento, Identificação, Análise Qualitativa, Análise Quantitativa, Planejamento de Respostas e Monitoramento e Controle). Nenhuma das ferramentas selecionadas fornece recursos específicos para suporte a esse gerenciamento. Todavia, o *dotProject* possui um módulo adicional (*Risks Management*) mas nem todos os aspectos são tratados por esse módulo. Segundo VIÉGAS (2010), são utilizados os fóruns e as listas de discussões para identificação dos riscos, já que isso é um processo interativo que envolve toda a equipe.

O módulo adicional de risco do *dotProject* trata resumidamente do cadastro dos registros referentes ao projeto e depende dos recursos adicionais (fóruns, repositório e listas de discussão) para auxílio na coleta de informações sobre os possíveis riscos do projeto. Da mesma maneira, através dos fóruns e lista de discussões é definida a análise quantitativa e qualitativa, entretanto não existe um campo especifico para definição de valores dessas análises. É importante

ressaltar que, apesar das limitações do módulo adicional do *dotProject*, ele possibilita o monitoramento de risco e o planejamento de resposta poderá ser realizado com adição de tarefas e, em última análise,possibilitando um controle e monitoramento de riscos.

Tabela 11. Resultado da análise da gerência de riscos.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Planejamento do gerenciamento de riscos	○	○	◐	○	○
Identificação de riscos	◐	◐	◐	◐	◐
Análise qualitativa de riscos	◐	◐	◐	◐	◐
Análise quantitativa de riscos	◐	◐	◐	◐	◐
Planejamento de respostas a riscos	◐	◐	◐	◐	◐
Monitoramento e controle de riscos	○	○	◐	○	○

4.4.9 Gerenciamento de Aquisições

Segundo o PMBOK, o planejamento de aquisições é um processo de documentação das decisões de compras, especificando a abordagem de identificação dos fornecedores em potencial. Assim, o processo é definido através da análise dos recursos existentes no projeto, e as informações fornecidas pelos relatórios podem auxiliar na tomada de decisões, e o planejamento das aquisições poderá ser realizado através do cadastrado de tarefas.

Observamos que o *]project-open[* em comparação às demais ferramentas, possibilita o planejamento de aquisições através do módulo de finanças, além de disponibilizar funcionalidades para o cadastro, edição e controle dos fornecedores relacionados ao projeto. Esses recursos não foram observados nas outras ferramentas. Entretanto, o planejamento das aquisições poderá ser realizada nas

demais ferramentas estudadas através da criação de projetos e cadastro das aquisições como tarefas a serem executadas.

Tabela 12. Resultado da análise da gerência de aquisições.

Processos	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Planejar compras e aquisições					
Planejar contrações					
Solicitar respostas de fornecedores					
Selecionar fornecedores					
Administração de contrato					
Encerramento de contrato					

4.5 ANÁLISE DOS RESULTADOS SOB A ÓPTICA DA GERÊNCIA DE PROJETOS

Uma vez levantados os recursos das ferramentas e feito o relacionamento entre os processos de gerência do PMBOK, elaborou-se a TABELA 13, para que as ferramentas pudessem ser comparadas de forma objetiva. Essa tabela apresenta o resumo com o número de recursos fornecidos pelas ferramentas de acordo com a simbologia adotada na TABELA 3.

Tabela 13. Numero de funcionalidades identificadas nas ferramentas.

Simbologia	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
	31	4	23	18	4
	11	21	17	20	22
	2	19	4	6	18

Observa-se que, o]*Project-open[* tem funcionalidades a mais que o *dotProject*. Isso se deve essencialmente aos módulos de recursos humanos e de finanças, uma vez que o *]Project-open[* possui recursos avançados nessas duas gerências. A versão disponível no *Source Forge* do *dotProject* não contém funcionalidades específicas para essas duas gerências. As duas ferramentas atendem diversos aspectos do gerenciamento de projetos, seja por funcionalidades nativas ou por meio de módulos adicionais, entretanto existem diferenças de recursos entre as ferramentas em determinadas áreas de aplicação, conforme a análise realizada.

No *EGroupware* identificaram-se recursos para gerenciamento de projetos, todavia, a maioria dos recursos visam o planejamento das tarefas e monitoramento de forma colaborativa, apresentando um número menor de funcionalidades para a gerência de risco, aquisições e custos. Não obstante, a ferramenta contém recursos para suporte à elaboração de cronogramas, gráfico de *Gantt* e geração de relatórios.

Os resultados demonstram que as ferramentas *Feng Office* e *TeamLab* contém menos recursos em comparação às demais, apresentando somente recursos básicos para o gerenciamento de projetos. Entretanto, não apresenta recursos avançados para controle e planejamento de um projeto, focando na interação e comunicação entre os colaboradores.

Diante dos resultados, é possível identificar em todas as ferramentas recursos básicos de gerência, destacando-se os projetos *dotProject* e *]Project-open[*, por possuírem um número maior de funcionalidades, segundo os processos de gerenciamento definido no PMBOK.

4.6 ANÁLISE DE FERRAMENTAS SOB A ÓPTICA DO SOFTWARE LIVRE

Para identificar nos projetos de ferramentas os recursos que atendessem os critérios quatro e cinco (deve possibilitar que uma atividade seja realizada de forma colaborativa e a obtenção de requisitos em qualquer fase do projeto), procurou-se funcionalidades que possibilitem, de forma colaborativa, a obtenção de

requisitos em qualquer etapa de desenvolvimento de um projeto de software, e que uma tarefa possa ser realizada por uma ou mais pessoas sem a necessidade de ser designada, pois segundo TAURION (2004 apud MARJORIE e PEREIRA, 2006), uma maneira de incentivar o desenvolvimento do software livre é permitir aos colaboradores a livre escolha das tarefas a serem desenvolvidas no projeto. Considerando-se essas características espera-se dos projetos de ferramentas as seguintes funcionalidades de colaboração entre participantes, segundo NEUS (2001) e REINHARDT (2003, apud PARREIRAS, 2004, p. 8):

- **Wiki**: Recursos que possibilitem qualquer usuário alterar o conteúdo de um documento, auxiliando na documentação do projeto;
- **Perfil dos colaboradores**: possibilite o armazenamento das informações e preferências de cada desenvolvedor;
- **Critérios para associação e inclusão**: possibilite que qualquer pessoa possa solicitar a inclusão de um projeto ou funcionalidade, a validade da solicitação deverá ser analisada por um membro da equipe;
- **Fórum**: possibilite um espaço compartilhado de conhecimento entre os usuários;
- **Lista de email**: Lista de usuários na qual desejam receber notificações sobre a atualização do projeto;
- **Chat**: meios que viabilize a troca de mensagens instantânea entre usuários;
- **Taxonomia**: possibilite ao usuário a navegação entre os projetos e identificando suas respectivas propriedades;
- **FAQS**[11]: Forneça uma lista com as dúvidas mais frequentes e suas respectivas respostas;
- **Repositório de falhas**: Forneça recursos que possibilite o relato de falhas identificadas no projeto;
- **Sistema de controle de versão**: Forneça recursos para controle das versões dos códigos-fonte desenvolvido pelos desenvolvedores;
- **Grupo de notícias**: possibilite o agrupamento das mensagens trocadas entre usuários de forma cronológica, ou por áreas de interesse de um determinado projeto.

[11] acrônimo inglês para perguntas mais frequentes (*Frequently Asked Question*).

É adotada a simbologia já descrita na TABELA 3 para melhor compreensão e entendimento, na análise comparativa das funcionalidades de cada ferramenta, em relação aos critérios de colaboração entre usuários definido no parágrafo anterior.

4.6.1 Recursos para edição e criação de documentos (Wiki)

Conforme a análise, não identificamos no repositório[12] do *dotProject*, módulo adicional para suporte a recursos do tipo *Wiki.* Segundo referências em fóruns e listas de discursões da comunidade do projeto, é recomenda a instalação separada desse recurso. Como exemplo, a comunidade do *dotProject* utilizada a ferramenta *MediaWiki*[13] para documentação do projeto. Em contra partida, a ferramenta *Feng Office* possui um ambiente colaborativo para criação e edição de documentos online, possibilitando de forma simples a documentação dos projetos através da interação dos usuários.

As ferramentas *EGroupware* e *TeamLab* possuem recursos nativos de Wiki, para colaboração da documentação. No caso do *TeamLab*, os recursos são similares ao *MediaWiki*, inclusive sua estruturação, com recursos avançados para inclusão de arquivo de imagem e formatação de texto.

No *]Project-open[*, o recurso *Wiki* é obtido através da instalação de três pacotes adicionais: *Content Management System* (responsável pela administração dos documentos criado), *Wiki* (Pacote principal, que fornece as funcionalidades para criação e edição dos documentos pelos colaboradores) e *intranet-wiki* (fornece componentes que proporcionam uma integração entre as páginas *wiki* e os objetos de negócio do *]Project-open[*).

4.6.2 Perfil dos colaboradores

Identificamos que todas as ferramentas possibilitam a criação de perfis para os colaboradores, em menor ou maior grau de informações. A ferramenta *Feng*

[12] Repositório do *dotProject* no *Source Forge*. Disponível em: < http://sourceforge.net/projects/dotmods/files/>.
[13] É software livre de código aberto, originalmente criado para a Wikipédia desenvolvido pela Wikimedia Foundation (MediaWiki, 2011).

Office possibilita o cadastro de endereços para as pessoas e das empresas/organizações às quais elas pertencem além da criação de relatórios sobre as atividades de uma pessoa, entretanto não existem recursos para determinar o desempenho e preferências do colaborador. No *dotProject* , os usuários cadastrados pelo administrador do sistema, de forma similar à demais ferramentas, permitem o agrupamento por empresa e departamento, além de possuir recursos para monitoramento das atividades de um colaborador, através da interface *user log.* Entretanto os perfis não são públicos, na maior parte, restrita ao administrador do sistema.De forma similar, o *]Project-open[* funcionalidades semelhantes ao *dotProject*.

Nas ferramentas *EGroupware* e *TeamLab*, o nível de customização da interface do colaborador contém um número maior de opções em comparação às demais, possibilitando a criação de perfis públicos. Entretanto, as ferramentas não possuem relatórios específicos de desempenho, e sim relatórios de atividades e interações.

4.6.3 Critérios para associação e inclusão

Conforme análise realizada nos processos da gerência de projetos, todas as ferramentas possibilitam a criação de um projeto e vinculado o nome de um responsável ao projeto. Entretanto, é necessário que seja definido um nível de permissão para a criação de um projeto e adição de tarefas. No *dotProject* essa configuração poderá ser realizada especificando o nível de permissão para cada módulos, com restrição somente para visualização dos registros, edição ,ou até mesmo a remoção de registros. A ferramenta]*Project-open[* funciona de forma similar, entretanto o nível de permissão é realizado através da definição do perfil do usuário (Funcionário, *Freelance*, Vendedor, Suporte, Clientes), nesse caso, o *Feng Office* segue o padrão de permissões do *dotProject* e *]Project-open[*.

Entre as ferramentas selecionadas, somente no *TeamLab* não foi possível identificar em suas configurações nível de acesso para os usuários de seu sistema. Assim, consideramos que essa ferramenta deve oferecer para qualquer usuário previamente cadastrado, a permissão para criação de um projeto ou nível de acesso para adição de uma tarefa. Não obstante, ressaltamos que o *TeamLab* contém um administrador, geralmente o responsável pela administração dos recursos

computacionais e cadastrados durante sua instalação, sendo também o responsável pela manutenção das contas dos usuários.

4.6.4 Fórum, Lista de email e Grupo de notícias

Devido à similaridade entres as funcionalidades e características dos recursos Fórum, lista de e-mails e grupo de notícias, foi realizado o estudo dessas funcionalidades em conjunto. Assim foi possível identificar em todas as ferramentas os recursos para o envio de notificações através de e-mails sobre as atualizações de um determinado projeto ou tarefa, sendo que a troca de mensagens sobre um determinado recurso ou discussão sobre algo relacionado ao projeto, só poderá ocorrer nos fóruns, sendo que o envio de email é usado somente para notificação. Na ferramenta *Feng Office* não foi identificada o recurso fórum.

4.6.5 Mensagens instantâneas

Entre as ferramentas analisadas, somente na ferramenta *TeamLab* foi identificada a funcionalidade para troca de mensagens instantâneas entre os usuários cadastrados. As demais ferramentas contêm somente os recursos mencionados anteriormente para interação entre os usuários cadastrados no sistema.

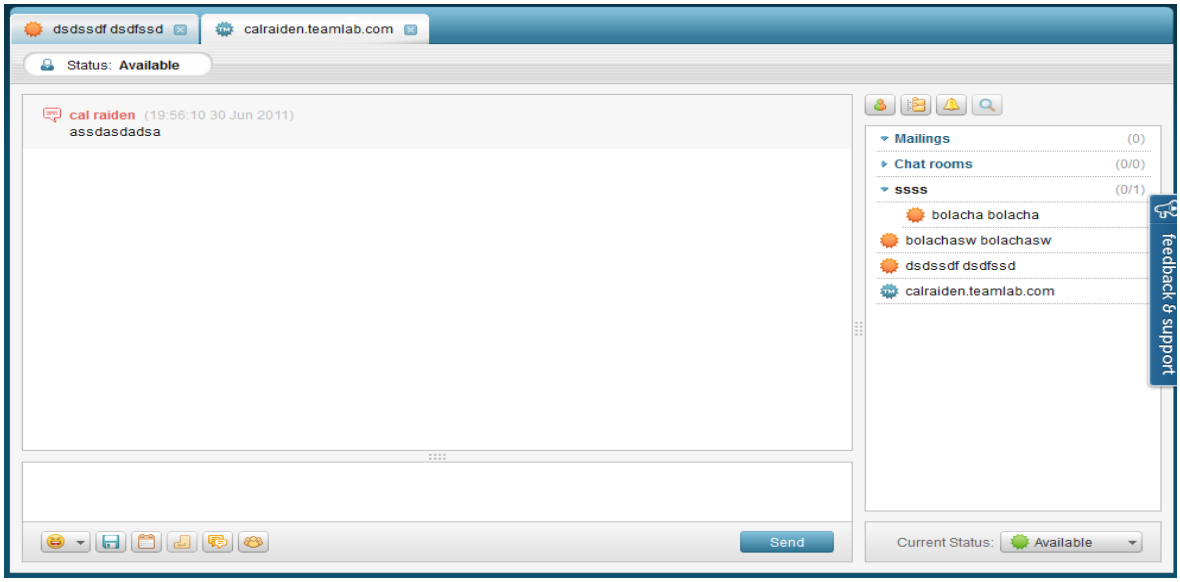

Figura 11. *Screenshot* do recurso para troca de mensagens instantâneas da ferramenta *TeamLab - Ascensio System*.

4.6.6 Taxonomia

Entre as ferramentas analisadas é possível identificar a possibilidade de navegação entre os projetos criados, em menor ou maior grau de informação sobre as propriedades. As ferramentas *dotProject* e *]Project-open[* são as que possibilitam um nível maior de navegação, pois permitem a criação de uma estrutura analítica entre as tarefas. A Taxonomia nesse caso é aplicada somente à categorização dos projetos, através de uma árvore de navegação, segundo PARREIRAS (2004). Nesse caso, não se aplica a um sentido mais amplo ou fora do escopo do estudo. Assim, todas as ferramentas possibilitam a navegação e fornecem informações sobre as propriedades dos projetos.

4.6.7 Dúvidas frequentes e suas respectivas respostas (FAQs)

Segundo PARREIRAS (2004), a lista de perguntas mais frequentes seguido de suas respostas tem como objetivo auxiliar no processo de aprendizado da utilização de uma ferramenta, buscando esclarecer as principais dúvidas dos usuários ou desenvolvedores. Baseado nesse contexto foi possível identificar em todas as ferramentas, além da documentação e manuais sobre o projeto, a possibilidade de criação de uma lista de perguntas mais frequentes e respostas. Essas informações se encontram disponível na página de projeto das ferramentas, alguma delas desenvolvidas de forma colaborativa. Assim, dentro do contexto de aprendizado das ferramentas, elas contêm o recurso que visam auxiliar seu aprendizado através do esclarecimento das perguntas identificadas com maior frequência ou relevância em seus fóruns, listas de discussão ou através da gestão do conhecimento dentro do projeto.

4.6.8 Repositório de Falhas

Conforme análise realizada anteriormente, o repositório de falhas corresponde de forma semelhante aos processos analisados no critério três (A ferramenta deve possibilitar a gerência de projetos), referente aos processos de controle integrado de mudanças da gerência de integração, e controle de mudanças do escopo da gerência do escopo. Dessa maneira, são definidos os mesmos valores resultantes da análise anterior para o recurso do repositório de falhas, uma vez que

as características são similares e as funcionalidades estão dentro do mesmo contexto analisado, visando o relato das falhas ocorrido dentro do projeto e o controle das correções realizadas.

4.6.9 Sistema de controle de versão

Segundo BROD e KÄFER (2009) e REIS (2003), os sistemas de controle de versão executam um importante papel dentro de um projeto de software livre, utilizados em grande uniformidade pelas comunidades existentes para controle do código fonte. Esse recurso é responsável por registrar as alterações propostas para o software, além de, permitir o desenvolvimento distribuído e sua publicação (ARAKAKI, 2008).

Entre as ferramentas estudadas não foi observada em nenhuma delas a disponibilização por padrão de recursos para o controle ou integração com sistemas de controle de versão para arquivos do projeto ou código fonte das aplicações. Entretanto, as ferramentas *dotProject* e *]Project-open[*, contêm módulos adicionais que possibilitam a integração com o Subversion1. No *]Project-open[* a integração é realizada através da inclusão do módulo *intranet-cvs-integration* e no *dotProject* através do módulo *Trac Integration*. A ferramenta *EGroupware* contém por padrão o modulo *Filemanager*, que apresenta características similares ao controle de versão de um arquivo, porém permite aos usuários gerenciar arquivos somente no servidor na qual o sistema foi instalado, possibilitando também o compartilhamento dos arquivos com outros usuários através de ACLs [14](*Access Control Lists),* sem versionamento desses arquivos.

Por fim, as ferramentas *TeamLab* e *FengOffice*, embora disponibilizem funcionalidades definidas como colaborativos, visando principalmente a criação, edição e compartilhamento de documentos entre seus usuários, não foi possível identificar nessas ferramentas recursos ou módulos adicionais para controle de versão.

[14] São listas de usuários e suas respectivas permissões, que define o controle de acesso ao um diretório ou arquivo.

4.7 ANÁLISE DOS RESULTADOS SOB A ÓPTICA DO SOFTWARE LIVRE

Uma vez levantados os recursos das ferramentas e feito o relacionamento entre as funcionalidades de colaboração das comunidades em software livre, segundo NEUS (2001) e REINHARDT (2003, apud PARREIRAS, 2004, p. 8), elaborou-se a TABELA 14 utilizando a simbologia adotada na TABELA 3.

Tabela 14. Funcionalidades de colaboração entre participantes.

Funcionalidades	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
Recursos colaborativos para a criação de documentos	●	◐	○	●	●
Perfil dos colaboradores	◐	◐	◐	●	●
Critérios para associação e inclusão.	◐	◐	◐	◐	●
Fórum	●	○	●	●	●
Lista de email	●	●	●	●	●
Grupo de notícias	●	●	●	●	●
Mensagens Instantâneas	○	○	○	○	●
Taxonomia de navegação entre projetos	●	●	●	●	●
Dúvidas frequentes e suas respectivas respostas	●	●	●	●	●
Repositório de falhas	●	◐	●	◐	◐
Sistema de controle de versões	●	○	●	◐	○

Para análise comparativa das ferramentas de forma objetiva, elaborou-se a tabela resumida abaixo.

Tabela 15. Números de funcionalidades de colaboração entre participantes identificadas.

Simbologia	]project-open[	Feng Office	dotProject	EGroupware	TeamLab
	8	4	7	7	9
	2	4	2	3	1
	1	3	2	1	1

A primeira ferramenta a ser destacada, o *TeamLab*, contém mais funcionalidades completas do que as demais, diferentemente da análise comparativa com os processos de gerência de projetos, isso deve essencialmente, por ser uma ferramenta desenvolvida inicialmente como plataforma de colaboração online, envolvendo recursos de mídia social (blog, fórum, wiki, favoritos), enquanto os recursos para o gerenciamento de projetos começaram a ser adicionados a partir de junho 2010. Observamos também que é possível identificar no *]Project-open[* recursos de colaboração online, embora tenha seu desenvolvimento focado na gerência de projetos. Isso se deve principalmente ao número de funcionalidades existentes na ferramenta, além dos módulos adicionais desenvolvidos pela comunidade desse projeto.

Os resultados demonstram que as ferramentas *dotProject* e *EGroupware* contém menos recursos de colaboração online em comparação as ferramentas *]Project-open[* e *TeamLab*, entretanto, a diferença no número de funcionalidades é pequena. Observamos que, com exceção do *Feng Office*, as ferramentas suportam funcionalidades características de colaboração entre participantes em uma comunidade de software livre, fornecendo suporte ao seu desenvolvimento.

Por fim, embora a ferramenta *Feng Office*, tenha sido categorizada pelo *Source Forge* como uma ferramenta para gerência de projetos, foi observado que suas funcionalidades são, em sua maioria, recursos para edição de documentos online e somente fornece recursos superficiais para o gerenciamento de projetos de desenvolvimento de software livre. Apesar de sua categorização, seus recursos não contemplam por completo o processo de desenvolvimento de um software livre nem atendem os processos de gerência de projetos conforme definido no PMBOK.

5. CONCLUSÕES

Neste capítulo são apresentadas as conclusões e a avaliação dos resultados do trabalho referente ao estudo realizado, contribuições do estudo e trabalhos futuros.

5.1 ESTUDO REALIZADO

O processo de desenvolvimento de um software livre difere do processo de desenvolvimento de um software proprietário, sendo realizado principalmente em ambientes colaborativos, dentro de grupos virtuais denominadas comunidades, formados por participantes geograficamente distribuídos. O Software Livre desenvolvido geralmente é proposto a partir de uma idéia pessoal ou necessidade de um participante da comunidade, e os demais participantes colaboram para o seu desenvolvimento. Por intermédio do estudo realizado, observou-se que o seu processo de desenvolvimento é auxiliado por ferramentas especificas, que visam à integração colaborativa dos participantes e a socialização do conhecimento.

O estudo realizado focou em realizar uma análise comparativa entre as características dos recursos das ferramentas e os processos da gerência de projetos, e os recursos de colaboração utilizados no desenvolvimento de um software livre. Foram selecionadas algumas ferramentas utilizadas para a gerência de projetos com maior relevância para as comunidades de software livre. Para realizar uma análise de forma objetiva, a seleção do grupo de ferramentas foi realizada utilizando-se diversos critérios definidos ao longo do estudo. Esses critérios tentam registrar as funcionalidades que atendam aos processos da gerência de projetos e forneçam suporte ao processo de desenvolvimento de um software livre. Além disso, observaram-se as principais características das ferramentas selecionadas e os recursos existentes.

Dentro desse contexto, e sob o ponto de vista da gerência de projetos, a maior parte das ferramentas selecionadas atende ou fornecem suporte em maior ou menor grau aos processos da gerência de projetos segundo o PMBOK. Alguns

processos não são atendidos por completo, devido à limitação de recursos ou necessidade que a tarefa que deve ser executada seja realizada por uma pessoa especifica. Em comparação com as características no processo de desenvolvimento de um software livre e os principais recursos utilizados para suporte ao seu desenvolvimento, observou-se que algumas ferramentas possuem recursos que dão suporte ao processo de desenvolvimento, devido a essas ferramentas atenderem aos critérios de integração entre participantes em um ambiente colaborativo.

Finalmente, os critérios utilizados no estudo demonstraram que maior parte das ferramentas analisadas, possuem recursos para atender aos processos da gerência de projetos, entretanto, as ferramentas com recursos colaborativos atendem em maior grau o processo de desenvolvimento de um software livre. Não obstante, todas as ferramentas contêm recursos que dão suporte ao planejamento e comunicação entre os participantes de um projeto em software livre.

5.2 CONTRIBUIÇÕES

Este trabalho oferece uma fonte de estudo para pesquisas que correlacionam Software Livre e Gerenciamento de Projetos, e principalmente uma análise comparativa entre algumas ferramentas de gerência de projetos existentes e seus respectivos recursos à luz das práticas das comunidades de Software livre bem como à luz das gerências de projetos descritas no PMBOK.

Os principais resultados e contribuições deste projeto são:

- Identificar ferramentas para gerência de projetos em repositórios públicos: A observação dos critérios de avaliação das ferramentas em um repositório público tornou possível a seleção das ferramentas com maior relevância nas comunidades de desenvolvimento de software livre.
- Identificar recursos nas ferramentas para suporte a Gerência de Projetos: O estudo das ferramentas e identificação dos seus recursos tornou possível a identificação das funcionalidades dessas ferramentas e correlacioná-las com os processos da gerência de projetos.
- Identificar recursos nas ferramentas para suporte ao desenvolvimento de um Software Livre: Durante o trabalho, foi possível levantar os principais

recursos utilizados pelas comunidades durante o desenvolvimento de um software livre, e realizar uma análise comparativa entre as funcionalidades existentes nas ferramentas estudadas.

5.3 TRABALHOS FUTUROS

O trabalho apresentou uma pesquisa sobre o modelo de desenvolvimento de um software livre e a gerência de projetos, além dos recursos existentes nas ferramentas dentro do contexto da gerência de projetos e software livre.

Assim, os principais trabalhos futuros são:

- Realização de uma pesquisa mais detalhada em relação a Gerência de Projetos e o Software Livre, fornecendo outras informações sobre as atividades relacionadas ao processo de gerência de projetos em projetos de software livre.
- Aprofundar o estudo sobre o processo de desenvolvimento de um Software Livre e a Gerência de Projetos, possibilitando a análise de outros aspectos e características no processo de um software livre, mas que não foram identificados nesse estudo.
- Relacionar mais detalhadamente as funcionalidades das ferramentas que obtiveram maior pontuação para suporte a Gerência de Projetos e o Software Livre, possibilitando a sugestão de melhorias em suas funcionalidades, de modo a atender o máximo possível as duas áreas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, Vicente M.; ALENCAR, Anderson F.; MACHADO, Murilo B.; EVANGELISTA, Rafael; SILVEIRA, Sergio A. **Software livre, cultura hacker e ecossistema da colaboração**. São Paulo/SP: Momento Editorial, 2009.

ARAKAKI, Alexandre Aguena. **Gerência de Projetos de Software Livre no Framework SAFE**. Dissertação de Mestrado do curso Ciência da Computação, Universidade Ferderal de Mato Grosso do Sul. Campo Grande/MS. 2008.

ARAÚJO, Marco A.; QUINTÃO, Patrícia L. **Ferramentas para Gerencia de Projetos**, Engenharia de Software Magazine, edição 25, ano 3. DevMedia Group, 2010.

BARBI, Fernando C., **Os 7 passos do gerenciamento de projetos**. Disponível em: <http://www.microsoft.com/brasil/msdn/Tecnologias/Carreira/GerencProjetos.mspx>. Acesso em: 18 de Out. 2010

ARANTES, Ednir et al. **Gerenciamento de Projetos**. São Paulo: Promom Bussiness e Technology Review, 2008. Disponível em:<http://www.promon.com.br/ portugues/ noticias /download/PBTR%20GE_para%20web.pdf>. Acesso em: 31 nov. de 2010.

BASTOS, A.; CAMEIRA, R. **Ferramentas de Apoio à Engenharia de Processos de Negócios**: Critérios de Classificação e Método de Análise de adequação a um projeto. In: ENEGEP, 20, out. 2000.

BARTIE, Alexandre. **Agilidade ou Controle Operacional? Os dois!.** Qualidade de Software: Entenda os principais conceitos sobre testes e Inspeção de Software, Rio de Janeiro/RJ, ano 1, n. 1,p. 22-28 ,2007.

BROD, Cesar A. de Azambuja; KÄFER, Joice. **Engenharia de Software para Software Livre**, Programa de Pós-Graduação – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre/RS, 2009.

CAMPOS, Lídio M. L.; LIMA, Alberto S**. Gerenciamento de Projetos de desenvolvimento de Software com o RUP e o PMBOK**, do Curso de Sistemas de Informação – Universidade Federal do Pará (UFPA) e Curso de Sistemas de Informação - Universidade Federal do Ceará. 2008.

CARDENAS, Yuri Gomes. **NODIPO**: Levantamento colaborativo de requisitos para desenvolvimento de software livre. Trabalho conclusão do curso Bacharel de Sistema de Informação, Universidade Federal de Santa Catarina. Florianópolis/SC, 2010.

CASTELLS, Manuel. **A Galáxia da Internet. Reflexões sobre a internet**, os negócios e a sociedade. Tradução: Maria Luiza X. de A. Borges. Rio de Janeiro: Jorge Zahar Ed., 2003.

CAVALCANTI, Dorval P.; CASTRO, Victor M. O. **Avaliação para Expansão do Uso de Software Livre no Superior Tribunal de Justiça** – STJ. Trabalho de Conclusão do curso de pós-graduação lato sensu em GERENCIAMENTO DE PROJETOS / PMBOK para a obtenção do grau de Especialista, Brasília/DF, 2007.

CISNEIROS, Hugo; JACOB, Moyses S.; MAZZA, Stelvio; PEREIRA, Tiago. **Engenharia de Software** - Modelo de Desenvolvimento Ágil SCRUM, do Curso Sistemas de Informação - Faculdade de Informática e Administração Paulista. 2009. 16 f. Artigo Acadêmico. São Paulo, 2009.

DIAS, André F. (2011). **O que é Gerência de Configuração**?. Disponível em:<http://www.pronus.eng.br/artigos_tutoriais/gerencia_configuracao/gerencia_configuracao.php>. Acesso em : 9 de jun. 2011.

dotProject - the Open Source Project Management tool. **dotProject**. Disponível em:<http://www.dotproject.net>. Acesso em: 18 de jun. 2011.

EARLEY, Erwin.(2010).**Understanding the Open-Source Quality Process**. Disponível em:<http://systeminetwork.com/article/understanding-open-source-quality-process>. Acesso em : 15 de jun. 2011.

EGroupware Enterprise Collaboration. **EGroupware Content Management** System. Disponível em:<http://www.egroupware.org>. Acesso em: 18 de jun. 2011.

EVANGELISTA, Rafael. (2006). **Software Livre x Open Source**. Disponível em:(http://wiki.softwarelivre.org/bin/view/CoberturaWiki/Pol%edticaELinguagemNosDebatesSobreOSoftwareLivre>. Acesso em: 17 set. 2010.

FALCÃO, Joaquim; JUNIOR, Tercio S. Ferraz; LEMOS, Ronaldo; MARANHÃO, Juliano; AFFONSO, Carlos P. Souza; SENNA, Eduardo; **Estudo sobre Software Livre**, Comissionado pelo Instituto Nacional de Tecnologia da Informação (ITI), Rio de Janeiro/RJ, 2005. Disponível em: <http://www.softwarelivre.gov.br/ publicacoes/

Estudo_FGV.pdf>. Acesso em: 22 de nov. 2010.

Feng Office - Your World Wide Office. **Feng Office**. Disponível em:<http://www.fengoffice.com>. Acesso em: 18 de jun. 2011.

FUKS, H.; RAPOSO, A.B.; GEROSA, M.A. **Engenharia de Groupware**: Desenvolvimento de Aplicações Colaborativas, in: Anais da XXI Jornada de Atualização em Informática, Capítulo 3, 2002. Disponível em: < http://www.les.inf. puc-rio.br/groupware >. Acesso em: 01 de nov. 2010.

FILHO, Edes Garcia da Costa; PENTEADO, Rosângela; SILVA, Júnia Coutinho Anacleto Silva; BRAGA, Rosana Teresinha V. (2005). **Padrões e Métodos Ágeis**: Agilidade no Processo de Desenvolvimento de Software. Disponível em: <subversion.assembla.com/svn/TCC_Agil/9673.pdf>. Acesso em 13 de jun. 2011.

FOWLER, Martin. (2003). **A Nova Metodologia**. Disponível em: <http://simplus.com. br/artigos/a-nova-metodologia/>. Acesso em: 13 de jun. 2011.

FREITAS, Celso C. C.; MOURA, Hermano P. **GMP**: Uma Ferramenta para a Gestão de Múltiplos Projetos. Artigo publicado no Simpósio Brasileiro de Sistemas de Informação - SBSI 2004.

GASPAR, Raphael Amato Salgado. **Aplicações do projeto descartes** , do Curso de Matemática do Centro de Ciências Matemáticas e da Natureza - UFRJ. 2008 p. 10. Monografia (Graduação em Matemática) - Universidade Federal do Rio de Janeiro, Rio de Janeiro/RJ, 2008.

KNIBERG, H. **Scrum e XP direto das Trincheiras**. InfoQ, 2007. Disponível em: < http://www.infoq.com>. Acesso em: 22 de nov. 2010.

Lindberg, Van., **Intellectual Property and Open Source**, 3ed, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2008 , **ISBN: 978-0-596-51796-0.**

Mantis. **Mantis Bug Tracker**. Disponível em <http://www.mantisbt.org/>. Acesso em: 24 de jun. 2011.

MARJORIE, Ivre R. M.; PEREIRA, Lúcio M., **Processo de Desenvolvimento de Software Livre**: Um Estudo de Caso do Projeto EAD Livre, Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais (PUC) Contagem/MGl, 2006.

MediaWiki. **Wikimedia Foundation**. Disponível em: < http://www.mediawiki. org/wiki/MediaWiki/pt>. Acesso em: 28 de jun. 2011.

MICHAELIS. **Moderno Dicionário Da Língua Portuguesa**. São Paulo: Melhoramentos, p. 203. 2009.

NARDI, Kleber. **PMBOK x SCRUM**: Como gerenciar um projeto de software? Disponível em: <http://www.pmkb.com.br/artigos-mainmenu-25/482-pmbok-x-scrum-como-gerenciar-um-projeto-de-software.html>. Acesso em: 5 de dez. 2010.

NUNES, Camila P.; STAA, Arndt V. **Processos de Software Open Source**, do Curso Ciência da Computação do departamento de informática - PUC. Monografia. PUC-RIO, Rio de Janeiro/RJ. 2007.

PACHECO, Guilherme Furtado. **Desenvolvimento de um software *touch screen*** para auxiliar na gerência do *taskboard,* do Curso de Ciência da Computação - UFSC. 2010. p. 15. Monografia (Graduação em Ciência da Computação) - Universidade Federal do Estado de Santa Catarina, Florianópolis/SC, 2010.

PAOLI, Stefano De ; D'ANDREA, Vincenzo. **How artefacts rule web-based communities**: practices of free software development. IJWBC 4(2): 199-219 (2007)

PARREIRAS, F. S.; SILVA, A. B. O.; BASTOS, J. S. Y.; BRANDÃO, W. C. **Informação e cooperação nas comunidades de desenvolvimento de software livre**: um panorama do cenário brasileiro. In: CINFORM, 5, 2004, Salvador. Anais... Salvador: UFBA, 2004.

PINTO, Licia de Cassia Nascimento. **Um modelo para apoiar a estruturação do conhecimento nas interações em comunidades virtuais de software livre**, do Curso de Tecnologia - UFRJ. 2009. p.41-45. Mestrado (Pós Graduação em Informática) - Universidade Federal do Rio de Janeiro,Rio de Janeiro/RJ, 2009.

PISKE, Otavio R.; CORRÊA, Ana P.; GREIN, Diego R. **Sistema de Controle de Versão** - Centro Universitário Positivo, 2005. Disponível em:<http://www. angusyoung. org /arquivos/artigos/sistemas_de_controle_de_versao.pdf>. Acesso em: 07 dez. de 2010.

PMBOK. **A Guide to the Project Management Body of Knowledge**. PMI-Project Management Institute. 3º Edição (2004), Newton Square, Pennsylvania, USA, 2004.

_____. **A Guide to the Project Management Body of Knowledge**. PMI-Project Management Institute. 4º Edição (2008), Newton Square, Pennsylvania, USA, 2008. ISBN: 978-1-933890-51-7

PMI. **Project management institute**, 2011. Disponível em: <http://www.pmi.org>. Acesso em: 31 jan. de 2011.

Project Open Online Documentation & Community . **]project-open[** - Project Management. Disponível em:<http://www.project-open.org>. Acesso em: 18 de jun. 2011.

OLIVEIRA, Fabio Braga. **Relações entre o PMBOK e metodologias ágeis de desenvolvimento de software**, do Curso de MBA em Gerenciamento de Projetos da Fundação Getúlio Vargas. 2008. 39 f. Monografia (Pós-Graduação lato sensu em Gerenciamento de Projetos) - Fundação Getúlio Varga, Campinas/SP, 2008.

RAYMOND, Eric S. A CATEDRAL E O BAZAR (**The Cathedral and the Bazaar**), 1998/99. Disponível em: < http://www.dominiopublico.gov.br/download/ texto/ tl000001.pdf >. Acesso em: 01 nov. 2011.

REIS, Christian R., **Caracterização de um Processo de Software para Projetos de Software Livre**, Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo para a obtenção do título de Mestre em Ciências da Computação e Matemática Computacional, São Carlos/SP, 2003.

OSI, Open Source Initiative. (2011). **The Open Source Definition** . Disponível em <http://www.opensource.org/docs/osd>. Acesso em: 12 de jun. 2011.

SAMPAIO, Marcio E**. Metodologia de Gerenciamento de Projetos**. Disponível em: <http://imasters.com.br/artigo/8392/gerencia/metodologia_de_gerenciamento_de_projetos/>. Acesso em: 01 de fev. de 2011.

SGANDERLA, Mauricio Andreazza; LACERDA, Guilherme Silva. **Melhorando a gerência e a construção de software com metodologias ágeis**, do Curso de Sistema de Informação do Centro Universitário Ritter dos Reis. 2010 p.5. Monografia (Graduação em Sistema de Informação) - UNIRITTER, Porto Alegre/RS,2010.

STALLMAN, Richard. (2001). **A GNU GPL e o American Way**. Disponível em:<http://biblioweb.sindominio.net/telematica/bib_Stallman_por.html>. Acesso em: 10 de set. 2010.

STEFANUTO, Giancarlo N.; DE LUCCA , José E.; ALVES, Angela M. ; FILHO , Sergio S. **Altec - XI Seminário Latino-Iberoamericano de Gestión Tecnológica**. O impacto Software Livre e de Código Aberto (SL/CA) nas Condições de Apropriabilidade na Indústria de Software Brasileira, Salvador, n. 1,out. 2005. Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments /_observatorio/ altec _apropriabilidade_sl.pdf>. Acesso em 01 de nov. 2010.

Subversion. **Apache Subversion**. Disponível em: <http://subversion.apache.org/ > . Acesso em : 1 de jul. 2011.

Sun Microsystems, **Open Source in the enterprise**: fulfilling the promise - white paper. Santa Clara/USA. 2009. Disponível em: < https://dct.sun.com/dct/forms/ reg_us_0708_838_0.jsp >. Acesso em: 10 de set. 2009.

TeamLab. **Ascensio System**. Disponível em:<http://www.teamlab.com/>. Acesso em: 19 de jun. 2011.

The Cyclops Group. **dotProject+**. Disponível em: <http://cyclops.telemedicina .ufsc. br/> . Acesso em: 24 de jun. 2011.

VIÉGAS, Diego Figueiredo; Moura, Eduardo; Guimarães, Paula A. Fuliana. **Avaliação das funcionalidades do sistema dotProject quanto às boas práticas descritas no guia PMBOK**. 2010. Trabalho de conclusão de Curso, do Curso de Especialização/MBA em Gerenciamento de Projetos de Software do Centro Universitário Euroamericano – Unieuro, Brasília/DF, 2010.

APÊNDICES

BOLACHA: ANÁLISE DE FERRAMENTAS PARA GERÊNCIA DE PROJETOS EM SOFTWARE LIVRE.

Claudionor Santos de Oliveira, Ronaldo José Abel.

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil

claudionors@inf.ufsc.br, ronaldo.abel@inf.ufsc.br

Abstract. *The objectives of this document are to accomplish an analysis was applied between the programs for support project management, correlating with resources used in development of free software, according to the literature review in this study.*

Resumo. *Este trabalho busca realizar uma análise comparativa entre ferramentas para gerência de projetos, correlacionando os seus recursos com os utilizado no desenvolvimento de um software livre, segundo a revisão bibliográfica realizada no trabalho.*

1. Introdução

Segundo a empresa Sun Microsystems (2009), o software livre ao longo dos anos tem crescido com o mesmo potencial que possui para ajudar tanto as empresas, como instituições e indivíduos, os quais buscam ferramentas que auxiliem em seus processos internos. Essa demanda crescente de desenvolvimento de software livre vem seguida de uma necessidade de planejamento para atender aos objetivos de quem procura ou requisita o software.

De acordo BROD e KÄFER (2009), muitas vezes, não existe um planejamento inicial, ou seja, a ausência de procedimentos definidos de coordenação, além da ausência de uma gerência, que garanta a qualidade do software. Segundo a American Society for Quality (2000, apud PMBOK, 2004, p. 194), podemos considerar qualidade de software, como o grau até o qual um conjunto de características inerentes satisfaz as necessidades e, segundo o PMBOK (2008), uma metodologia de gerência, define as abordagens ferramentas e fontes de dados que podem ser usadas para realizar o gerenciamento de um projeto.

Portanto, o grande desafio é fundir os conceitos das metodologias de gerência de projetos com a visão do desenvolvimento em software livre, diante de suas características. No contexto desse artigo, é realizando uma análise comparativa sobre um grupo de ferramentas para gerência de projetos, de tal modo que seja possível correlacioná-las com as características identificadas do modelo de desenvolvimento de um software livre.

2. Gerência de Projetos

A Gerência de Projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos (PMBOK, 2008) ou objetivos pré-definidos, num certo prazo, com um custo estabelecido e metas de qualidades a serem cumpridas, através da mobilização de recursos

financeiros, tecnológicos e humanos. Segundo ARANTES et al. (2008), por ter, obrigatoriamente, início e fim definidos, os projetos se diferenciam de operações contínuas. Essa característica implica que são iniciados, durante seu processo, evoluem e, por fim, são finalizados. Essas temporalidades dos projetos exigem a adoção de um ciclo de vida, similar ao ciclo de vida de um software, constituído de fases elaboradas com o planejamento das atividades a serem realizadas.

De acordo com o PMBOK (2008), os processos relacionados às suas atividades a serem realizadas durante a execução do projeto podem ser inseridos em grupos, e difundidos em áreas de conhecimento (Gerenciamento de Integração, Gerenciamento de Escopo, Gerenciamento do Tempo, Gerenciamento de Custo, Gerenciamento da Qualidade, Gerenciamento de Recursos Humanos, Gerenciamento de Comunicações, Gerenciamento de Risco, Gerenciamento de Aquisições), os grupos de processo podem ser definidos como **Iniciação** (fase inicial de todo projeto, onde ao final da fase é obtido um documento formal representando o projeto e seu objetivo), **Planejamento** (fase responsável por detalhar tudo que será realizado, incluindo prazo, custos, atividades, recursos financeiros e humanos, etc.), **Execução** (a fase de execução, produção e materialização do que foi planejado nas fases anteriores.), **Monitoramento e controle** (fase de acompanhamento do projeto, revisão e ajuste das atividades de acordo com o progresso e desempenho do projeto, podendo ocorrer em paralelo ou isoladamente das demais fases.) e **Encerramento** (última fase do projeto, responsável pela avaliação, homologação das atividades executadas e pacotes de trabalhos entregues, além da atualização da documentação e aprendizados ocorridos durante as fases do projeto.).

De maneira geral, uma gerência de projetos é desenvolvida por uma metodologia, composta por técnicas e processos que visam a aumentar e garantir a eficiência do trabalho realizado dentro de uma organização (SAMPAIO, 2008), através de metas ou estruturas necessárias para que um processo de desenvolvimento apresente melhorias na qualidade no processo de desenvolvimento de um software (TAMAKI, 2007, apud CAMPOS e LIMA, 2008). Uma metodologia de gerência de projetos pode ser baseada em modelos tradicionais da engenharia de software ou metodologias ágeis, o modelo tradicional tenta prever as mudanças durante a fase de planejamento, visando controlas durante todo o processo de desenvolvimento do software (TAMAKI, 2007, apud CAMPOS e LIMA, 2008) e as metodologias ágeis se diferem por ser adaptativas as mudança de mercado ou requisitos ao invés de predeterminantes, além de orientados a pessoas ao invés de serem orientados a processos (FOWLER, 2003) .Portando a diferença entre as metodologias tradicionais e ágeis está na maneira de como as mudanças são tratadas durante o projeto.

Por fim, uma ferramenta de gerência de projetos consiste da agregação dos processos, ferramentas, técnicas, metodologias, recursos e procedimentos para o gerenciamento de um projeto (PMBOK, 2008, p. 322), apesar das ferramentas para gerência de projetos apresentarem similaridades entre si, seus processos variam de acordo com a metodologia adotada, focam em áreas especificas de uma gerência ou metodologia (FREITAS e MOURA, 2004).

3. Software Livre

Podemos definir de maneira simplista um software livre, como uma questão de liberdade, e não de preço, uma que qualquer pessoa tem a liberdade de estudar, mudar e redistribuir o software, segundo STALLMAN (2001), assim compreende-se como software livre aquele que permite aos seus usuários os quatros princípios definido pela FSF (Free Software Foundation) de direito ou liberdade (FALCÃO et al.,2005):

1. Executar o software;
2. Estudar o software;
3. Redistribuir cópias;
4. Aperfeiçoar o software.

Um software livre pode ser também ser considerado como um movimento tecnológico em um ambiente colaborativo, pois é seu principal meio de comunicação e interação entre os indivíduos que o produz (CASTELLS, 2003). Esse ambiente de colaboração é denominado **comunidade**, o coração do software livre, e os indivíduos que a compõem e participam em comum consenso nos projetos relacionados à sua filosofia (AGUIAR et al., 2009). Por esses fatores de colaboração e questões de liberdade, que o software livre demonstra as divergências entre o processo de desenvolvimento de um software livre e de um software tradicional (PAOLI e ANDREA, 2006). Essas divergências podem ser melhor compreendida através da metáfora da Catedral e o Bazar, criada por RAYMOND (1999), onde a **Catedral** corresponde a um processo de desenvolvimento unilateral ou tradicional, comparando a um pastor que entrega sua mensagem aos fiéis sem ser contestado, e o **Bazar**, corresponde a um ambiente de troca de ideias (BROD e KÄFER, 2009, p. 8).

Outro aspecto importante no desenvolvimento do software livre em comparação ao modelo de software tradicional, e a coleta de requisitos, pois é menos rígida devido ao fato de que grande parte dos projetos surge de motivações pessoais e replicam funcionalidades de algum produto existente (BROD e KÄFER, 2009 p. 9), além que, que a coleta dos requisitos é realizada de forma continua e coletiva nas comunidades (GASSER et al., 2003 apud CARDENAS, 2010, p. 20). Entretanto, de forma similar ao desenvolvimento de um software tradicional e as metodologias para gerência de projetos, o processo de desenvolvimento de um software livre poder ser definido através de um conjunto de atividades ou processos, segundo SOMMERVILLE (1995, apud PINTO, 2009), essas atividades variam segundo a filosofia da comunidade.

É importante ressaltar que no desenvolvimento de um software livre, poucas ferramentas são utilizadas e grande uniformidade entre os projetos de quais são aplicadas e as ferramentas para controle de versões (CVS) possuem popularidade (REIS, 2003 p. 48), os CVS exercem um papel fundamental dentro das comunidades em software livre (BROD e KÄFER, 2009), pois é um dos principais fatores para melhora de qualidade e produtividade, além de permite um controle do desenvolvimento distribuído, bem como exposição do código a um imenso número de pessoas.

Apesar das divergências entre o modelo de desenvolvimento proprietário ou tradicional de software, segundo ARAKAKI (2008), é possível afirmar que muitas comunidades possuem um gerenciamento de projetos, de forma similar ao modelo de desenvolvimento do software proprietário, esse gerenciamento de projetos visam principalmente a

conversão do conhecimento para suporte e compartilhamento conhecimento adquirido através da interação entre os indivíduos que compõem a comunidade (RUDZI e JONSON, 2003, apud PINTO, 2009). Dessa forma, podemos concluir que praticas de gerenciamento de projetos nas comunidades e no desenvolvimento do software livre.

Finalmente, segundo ARAKAKI (2008), a gerência de coordenação das comunidades tem uma forte relação com todas as áreas de conhecimento do PMBOK, apesar dos projetos de software livre não apresentarem uma estrutura formal, as suas gerências são voltadas a comunicação e qualidade. Implicando que todas as gerências do software livre têm ligação com a gerência de qualidade e comunicação do PMBOK.

4. Ferramentas de Gerência de Projetos

Segundo MARJORIE e PERREIRA (2006), a adoção de ferramentas durante o ciclo de desenvolvimento de um software, pode conduzir a um sistema com menor número de erros e maior grau de qualidade. A aplicação correta de ferramentas, alinhada ao conhecimento podem aumentar as chances de um projeto alcançar seu objetivo (CAVALCANTI e CASTRO, 2007, p. 13). De acordo com ARAUJO e QUINTÃO (2010), conhecer ferramentas de gerenciamento de projetos é cada vez mais importante na medida em que uma organização avança em seu nível de maturidade.

A maioria das ferramentas para gerência de projetos apresenta similaridade em suas funcionalidades, pois foram desenvolvidas para atender os projetos de propósito em geral, segundo FREITAS e MOURA (2004), entretanto, algumas focam em áreas especificas como gerência de custo, gerência de atividades, comunicação e integração da equipe, entre outras áreas.

À medida que uma organização avançar em seu nível de maturidade, é cada vez mais importante conhecer as ferramentas para gerenciamento de projetos existente, segundo ARAUJO e QUINTÃO (2010). Não obstante, o desenvolvimento do software livre, se difere do modelo tradicional ou proprietário, assim faz necessário à definição de critérios para auxiliar na seleção das ferramentas para o seu desenvolvimento, além disso, a definição dos critérios viabiliza a análise e comparação das ferramentas.

Baseado no contexto da revisão bibliográfica realizada, o primeiro critério define que uma ferramenta para gerência de projetos para suporte ao desenvolvimento de um software livre deve ter seu código-fonte aberto disponível para análise e possíveis propostas de modificações de acordo com os princípios do software livre.

O segundo critério define que a ferramenta deve apresentar uma interface para acesso através da Web aos usuários, pois o modelo de desenvolvimento de um software livre, em sua maioria ocorre em comunidades distribuídas geograficamente, tendo como o principal meio de comunicação, interação e publicação através da Web (BROD e KÄFER, 2009).

Os dois primeiros critérios, essencialmente, são critérios de filtragem das ferramentas para gerência de projetos, dessa forma foi realizada inicialmente uma pesquisa exploratória no *Source Forge,* aplicando os dois critérios. O repositório publico S*ource Forge,* foi escolhido como base de pesquisa, por ser um ambiente de desenvolvimento colaborativo utilizado por diversas comunidades (ARAKAKI,

2008, p. 19). Entretanto, devido os critérios de filtragem e seleção adotada incialmente, obtivermos como resultado em torno de 4298 projetos de ferramentas para gerência de projetos, aplicando em seguida o critério de relevância dos projetos para as comunidades, de acordo com a classificação do próprio *Source Forge*, e por fim, baseando-se no grau de relevância dessas ferramentas em comparação as demais, segundo informações do *Source Forge*, cinco projetos de ferramentas foram selecionados.

O primeiro projeto a ser selecionado foi o sistema **]Project-open[**, uma aplicação web, com recursos para gerenciamento de projetos e ERP(*Enterprise Resource Planning*), o sistema **Feng Office** projetada na Universidad de la República do Uruguai, por Conrado Viña, como trabalho de conclusão de curso, o sistema **dotProject**, iniciado em 2000 por dotmarketing.org e atualmente é mantida por sua própria comunidade, tendo como principais desenvolvedores Adam Donnison, Karen Chisholm, Gregor Erhardt, Ivan Peevski, Eamon Brosnan e Benjamin Young. Por fim, o quarto projeto de ferramentas a ser selecionado foi o sistema **eGroupware,** iniciado em 2000, com desenvolvimento do projeto phpgroupware, que foi baseado no webdistro, separando-se em 2003 com intuito de ter um caráter mais comunitário no seu desenvolvido em comparação aos seus antecessores, e o quinto projeto foi o sistema **TeamLab**, desenvolvida pela empresa *Ascensio System* e licenciado sob a licença GNU GPLv3, iniciado em dezembro de 2009 como uma simples plataforma de colaboração online, na qual abrangia recursos de mídia social (blog, fórum, wiki, favoritos).

Segundo ARAKAKI (2008), é possível identificar nos projetos de software livre, padrões de gerenciamento e estrutura de controle, possibilitando o correlacionamento das praticas de gerência realizado nas comunidades de software livre com as nove gerências do PMBOK, que caracterizam os principais aspectos envolvidos em um projeto e no seu gerenciamento, de acordo com o PMI. Dessa forma, realizando o levantamento das informações, no primeiro momento através de estudos acadêmicos existente e no segundo momento um estudo direto, para descoberta dos recursos presentes nas ferramentas que fornecesse suporte as áreas de conhecimento e processos, foi possível identificar que em todas as ferramentas possuem recursos básicos de gerência, destacando-se os projetos *dotProject* e *]Project-open[*, por possuírem um número maior de funcionalidades.

De acordo com NUNES e STAA (2007), o desenvolvimento de um software livre na maioria é realizado pelo interesse no compartilhamento do conhecimento, podendo um individuo interagir em diversas áreas de conhecimento em um projeto, com maior aptidão ou interesse. Segundo MARJORIE e PERREIRA (2006), a maioria dos projetos em software livre não possui uma especificação de requisitos formal nos moldes da engenharia de requisitos tradicional. Sendo os colaboradores as principais fontes de obtenção dos requisitos a medida que vai sendo desenvolvido, segundo NUNES e STAA (2007, p.8). Uma vez, identificado essas características, no desenvolvimento de um software livre, além de atender os critérios da gerência de projetos, as ferramentas devem possibilitar que uma atividade seja realizada de forma colaborativa e a obtenção de requisitos em qualquer fase do projeto. Baseado

nessas características espera-se dos projetos de ferramentas, as seguintes funcionalidades de colaboração entre participantes, segundo NEUS (2001) e REINHARDT (2003, apud PARREIRAS, 2004, p. 8):

1. Wiki: Recursos que possibilitem a criação e edição de documentos online;
2. Perfil dos colaboradores: possibilite o armazenamento das informações de cada colaborador;
3. Critérios para associação e inclusão: possibilite que qualquer pessoa possa solicitar a inclusão de um projeto ou funcionalidade;
4. Fórum: possibilite um espaço compartilhado de conhecimento entre os usuários;
5. Lista de email: Envio de notificações por email para grupos de usuários;
6. Chat: meios que viabilize a troca de mensagens instantânea;
7. Taxonomia: possibilite ao usuário a navegação entre os projetos;
8. FAQS : lista com as dúvidas mais frequentes e suas respectivas respostas;
9. Repositório de falhas: recursos para o relato de falhas;
10. Sistema de controle de versão: recursos para controle das versões dos códigos-fonte;
11. Grupo de notícias: possibilite o agrupamento das mensagens trocadas entre usuários.

Uma vez levantada os recursos das ferramentas e relacionadas entre as funcionalidades de colaboração entre participantes das comunidades em software livre, observa-se uma pequena diferença no número de funcionalidades entre os sistemas analisados. Entretanto, apesar da pouca diferença, a ferramenta *TeamLab* contém um número maior de funcionalidades, para colaboração online em comparação as demais , embora atenda um número reduzido de processos para uma gerência de projetos. A ferramenta *EGroupware*, atende de forma equilibrada, porém com menos funcionalidade que as três primeiras ferramentas. Finalmente, a ferramenta *Feng Office*, contém em sua maioria, funcionalidades voltadas à edição e criação de documentos online, fornecendo poucos recursos que atendam tanto a gerência de projetos como o desenvolvimento de um software livre.

5. Conclusões

Dentro contexto do estudo realizado, foi observada maior parte das ferramentas selecionadas, atende ou fornecem suporte aos processos da gerência de projetos segundo o PMBOK, sendo que alguns processos não são atendidos por completo, devido à limitação de recursos, ou necessidade que a tarefa a ser executada, seja realizada por uma pessoa especifica. Entretanto, observa-se também que as ferramentas focam em uma determinada área do conhecimento, e por isso, não visam atender todos os recursos esperado em uma ferramenta que forneça suporte a gerência de projetos.

Em comparação as características do processo de desenvolvimento de um software livre, e os principais recursos utilizados para suporte ao seu desenvolvimento, observaram-se que algumas ferramentas possuem recursos que fornece suporte ao eu desenvolvimento, devido essas ferramentas atenderem aos critérios de integração entre participantes em um ambiente colaborativo.

Finalmente, os critérios

apresentados no estudo, demonstraram que embora uma ferramenta fornecesse recursos que visam atender aos processos da gerência de projetos, não fornecem recursos por completo para o processo de desenvolvimento de um software livre. Não obstante, contem recursos que dão suporte ao planejamento e comunicação entre os participantes de um projeto em software livre.

7. References

AGUIAR, Vicente M.; ALENCAR, Anderson F.; MACHADO, Murilo B.; EVANGELISTA, Rafael; SILVEIRA, Sergio A. Software livre, cultura hacker e ecossistema da colaboração. São Paulo/SP: Momento Editorial, 2009.

ARAKAKI, Alexandre Aguena. Gerência de Projetos de Software Livre no Framework SAFE. Dissertação de Mestrado do curso Ciência da Computação, Universidade Ferderal de Mato Grosso do Sul. Campo Grande/MS. 2008.

ARAÚJO, Marco A.; QUINTÃO, Patrícia L. Ferramentas para Gerencia de Projetos, Engenharia de Software Magazine, edição 25, ano 3. DevMedia Group, 2010.

ARANTES, Ednir et al. Gerenciamento de Projetos. São Paulo: Promom Bussiness e Technology Review, 2008. Disponível em:<http://www.promon.com.br/ portugues/ noticias /download/PBTR%20GE_para%20web.pdf>. Acesso em: 31 nov. de 2010.

BROD, Cesar A. de Azambuja; KÄFER, Joice. Engenharia de Software para Software Livre, Programa de Pós-Graduação – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre/RS, 2009.

CAMPOS, Lídio M. L.; LIMA, Alberto S. Gerenciamento de Projetos de desenvolvimento de Software com o RUP e o PMBOK, do Curso de Sistemas de Informação – Universidade Federal do Pará (UFPA) e Curso de Sistemas de Informação - Universidade Federal do Ceará. 2008.

CARDENAS, Yuri Gomes. NODIPO: Levantamento colaborativo de requisitos para

desenvolvimento de software livre. Trabalho conclusão do curso Bacharel de

Sistema de Informação, Universidade Federal de Santa Catarina. Florianópolis/SC,

2010.

CASTELLS, Manuel. A Galáxia da Internet. Reflexões sobre a internet, os negócios e a sociedade. Tradução: Maria Luiza X. de A. Borges. Rio de Janeiro: Jorge Zahar Ed., 2003.

CAVALCANTI, Dorval P.; CASTRO, Victor M. O. Avaliação para Expansão do Uso de Software Livre no Superior Tribunal de Justiça – STJ. Trabalho de Conclusão do curso de pós-graduação lato sensu em GERENCIAMENTO DE PROJETOS / PMBOK para a obtenção do grau de Especialista, Brasília/DF, 2007.

dotProject - the Open Source Project Management tool. dotProject. Disponível em:<http://www.dotproject.net>. Acesso em: 18 de jun. 2011.

EGroupware Enterprise Collaboration. EGroupware Content Management System. Disponível em:<http://www.egroupware.org>. Acesso em: 18 de jun. 2011.

FALCÃO, Joaquim; JUNIOR, Tercio S. Ferraz; LEMOS, Ronaldo;

MARANHÃO, Juliano; AFFONSO, Carlos P. Souza; SENNA, Eduardo; Estudo sobre Software Livre, Comissionado pelo Instituto Nacional de Tecnologia da Informação (ITI), Rio de Janeiro/RJ, 2005. Disponível em: <http://www.softwarelivre.gov.br/ publicacoes/ Estudo_FGV.pdf>. Acesso em: 22 de nov. 2010.

Feng Office - Your World Wide Office. Feng Office. Disponível em:<http://www.fengoffice.com>. Acesso em: 18 de jun. 2011.

FOWLER, Martin. (2003). A Nova Metodologia. Disponível em: <http://simplus.com. br/artigos/a-nova-metodologia/>. Acesso em: 13 de jun. 2011.

FREITAS, Celso C. C.; MOURA, Hermano P. GMP: Uma Ferramenta para a Gestão de Múltiplos Projetos. Artigo publicado no Simpósio Brasileiro de Sistemas de Informação - SBSI 2004.

MARJORIE, Ivre R. M.; PEREIRA, Lúcio M., Processo de Desenvolvimento de Software Livre: Um Estudo de Caso do Projeto EAD Livre, Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais (PUC) Contagem/MGl, 2006.

NUNES, Camila P.; STAA, Arndt V. Processos de Software Open Source, do Curso Ciência da Computação do departamento de informática - PUC. Monografia. PUC-RIO, Rio de Janeiro/RJ. 2007.

PAOLI, Stefano De ; D'ANDREA, Vincenzo. How artefacts rule web-based communities: practices of free software development. IJWBC 4(2): 199-219 (2007).

PARREIRAS, F. S.; SILVA, A. B. O.; BASTOS, J. S. Y.; BRANDÃO, W. C. Informação e cooperação nas comunidades de desenvolvimento de software livre: um panorama do cenário brasileiro. In: CINFORM, 5, 2004, Salvador. Anais... Salvador: UFBA, 2004.

PINTO, Licia de Cassia Nascimento. Um modelo para apoiar a estruturação do conhecimento nas interações em comunidades virtuais de software livre, do Curso de Tecnologia - UFRJ. 2009. p.41-45. Mestrado (Pós Graduação em Informática) - Universidade Federal do Rio de Janeiro, Rio de Janeiro/RJ, 2009.

PMBOK. A Guide to the Project Management Body of Knowledge. PMI-Project Management Institute. 3º Edição (2004), Newton Square, Pennsylvania, USA, 2004.

_____. A Guide to the Project Management Body of Knowledge. PMI-Project Management Institute. 4º Edição (2008), Newton Square, Pennsylvania, USA, 2008. ISBN: 978-1-933890-51-7.

PMI. Project management institute, 2011. Disponível em: <http://www.pmi.org>. Acesso em: 31 jan. de 2011.

Project Open Online Documentation & Community .]project-open[- Project Management. Disponível em:<http://www.project-open.org>. Acesso em: 18 de jun. 2011.

RAYMOND, Eric S. A CATEDRAL E O BAZAR (The Cathedral and the Bazaar), 1998/99. Disponível em: < http://www.dominiopublico.gov.br/download/ texto/ tl000001.pdf >. Acesso em: 01 nov. 2011.

REIS, Christian R., Caracterização de um Processo de Software para Projetos de Software Livre, Dissertação apresentada ao Instituto de Ciências

Matemáticas e de Computação da Universidade de São Paulo para a obtenção do título de Mestre em Ciências da Computação e Matemática Computacional, São Carlos/SP, 2003.

SAMPAIO, Marcio E. Metodologia de Gerenciamento de Projetos. Disponível em: <http://imasters.com.br/artigo/8392/gerencia/metodologia_de_gerenciamento_de_projetos/>. Acesso em: 01 de fev. de 2011.

STALLMAN, Richard. (2001). A GNU GPL e o American Way. Disponível em:<http://biblioweb.sindominio.net/telematica/bib_Stallman_por.html>. Acesso em: 10 de set. 2010.

STEFANUTO, Giancarlo N.; DE LUCCA , José E.; ALVES, Angela M. ; FILHO , Sergio S. Altec - XI Seminário Latino-Iberoamericano de Gestión Tecnológica. O impacto Software Livre e de Código Aberto (SL/CA) nas Condições de Apropriabilidade na Indústria de Software Brasileira, Salvador, n. 1,out. 2005. Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments /_observatorio/ altec _apropriabilidade_sl.pdf>. Acesso em 01 de nov. 2010.

Subversion. Apache Subversion. Disponível em: <http://subversion.apache.org/ > . Acesso em : 1 de jul. 2011.

Sun Microsystems, Open Source in the enterprise: fulfilling the promise - white paper. Santa Clara/USA. 2009. Disponível em: < https://dct.sun.com/dct/forms/ reg_us_0708_838_0.jsp >. Acesso em: 10 de set. 2009.

TeamLab. Ascensio System. Disponível em:<http://www.teamlab.com/>. Acesso em: 19 de jun. 2011.