

Integrando *Groupware* e Gerenciamento de Requisitos no Suporte à Engenharia de Requisitos Distribuída

Regiane Andrade Brito^{1,2}, Alexandre Marcos Lins de Vasconcelos¹
¹*Centro de Informática – Universidade Federal de Pernambuco (UFPE)*
Caixa Postal 7851, 50732-970, Recife – PE – Brasil
²*Serviço Federal de Processamento de Dados (SERPRO)*
Av. Perimetral da Ciência, 2010, 66077-530, Belém – PA – Brasil
{rab3, amlv}@cin.ufpe.br

Abstract

Esse artigo propõe a integração de groupware e gerenciamento de requisitos para permitir o registro e manutenção de rationale sobre requisitos e dessa forma, aumentar a compreensão destes em Desenvolvimento Distribuído de Software (DDS), além de ampliar a coordenação e comunicação entre os participantes. Para isso, uma análise de ferramentas de gerenciamento de requisitos existentes foi realizada para verificar a viabilidade da proposta e um protótipo foi desenvolvido e integrado a um groupware open source existente. Um estudo de caso é apresentado para validar a proposta.

1. Introdução

O desenvolvimento de software com equipes geograficamente distribuídas é uma realidade e várias empresas já utilizam esse estilo de desenvolvimento, denominado Desenvolvimento Distribuído de Software (DDS) [1;2]. Em um outro cenário está o desenvolvimento de software livre, onde um grupo de pessoas colabora para o aprimoramento contínuo de produtos de software cujo código fonte e demais artefatos relacionados encontram-se disponíveis através da internet. Vale ressaltar que o desenvolvimento de software já é uma atividade complexa quando realizada localmente, com equipes distribuídas essa complexidade é ainda maior.

Em ambos os cenários é necessário vencer as barreiras impostas pela distância geográfica. Estudos indicam que equipes distribuídas enfrentam problemas como: falta de coordenação, comunicação ineficiente, diferenças de fusos horários, diferenças culturais e perda do espírito de equipe.

Nesse contexto, a disciplina de engenharia de requisitos merece maior atenção [3], por ser considerada a maior responsável pela qualidade do sistema final [4] e por ser uma atividade complexa que requer muita interação entre pessoas com

conhecimentos e culturas diferentes, o que tende a ser ampliado nessas situações de distribuição.

Engenharia de Requisitos é um processo extremamente relacionado à aquisição e representação de informações, no entanto a maioria das ferramentas de engenharia de requisitos fornece pouco ou nenhum suporte à colaboração. Para reduzir o problema da falta de compreensão e gerenciamento de informações sobre requisitos em DDS é proposta a integração de *groupware* e ferramentas de gerenciamento e requisitos.

Uma aplicação foi desenvolvida e incorporada a um *groupware* bem conhecido no âmbito de software livre, o eGroupware. Juntos, a aplicação desenvolvida e o ambiente eGroupware possibilitam que o suporte efetivo à colaboração durante a etapa de engenharia de requisitos com equipes distribuídas seja possível.

A seção 2 apresenta uma discussão sobre engenharia de requisitos com equipes distribuídas. A seção 3 avalia ferramentas de gerenciamento de requisitos quanto o suporte necessário para equipes distribuídas. A seção 4 apresenta a ferramenta CODIPSE-Req construída para validar a proposta. A seção 5 apresenta um estudo de caso. A seção 6 apresenta as considerações finais e a seção 7, as referências.

2. Engenharia de Requisitos com equipes distribuídas

Engenharia de requisitos (ER) é o processo de descoberta de funcionalidades e restrições de um sistema, identificando stakeholders e suas necessidades, bem como documentando essas descobertas de uma forma que possibilite a análise, comunicação e implementação futura [4].

No desenvolvimento com equipes co-localizadas, a coleta de requisitos ocorre em reuniões face a face, considerada a forma mais efetiva de comunicação [5]. Esses encontros tornam mais fáceis também as atividades de negociação e resolução de conflitos. Em DDS essas atividades tendem a ser dificultadas por problemas inerentes da distribuição geográfica, além

dos problemas já conhecidos como dificuldade de comunicação entre os *stakeholders* [6]. Alguns estudos de caso relatam experiências, apresentando as lições aprendidas [7;8].

Existem algumas propostas na literatura que também fornecem suporte ao trabalho colaborativo em engenharia de requisitos, como o ambiente CRETA [9] que é baseado na web e permite a integração com um processo de engenharia de requisitos e o Teamwave instanciado para engenharia de requisitos [5]. No

entanto, esses ambientes não tratam diretamente sobre desenvolvimento com equipes distribuídas e não mencionam como o rastreamento de requisitos deve ser tratado.

Nesta proposta, os participantes estão totalmente distribuídos conforme mostrado na figura 1 e usam um ambiente adequado para trabalhar cooperativamente. Na figura 1.b), todos os usuários (engenheiro de requisitos, clientes, entre outros) estão representados de forma genérica como “*stakeholder*”.

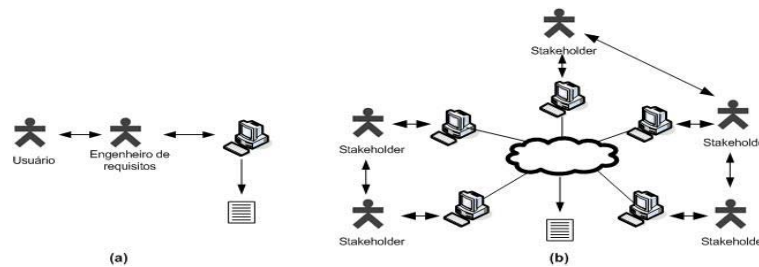


Figura 1 - (a) processo mediado por um desenvolvedor (b) um processo totalmente cooperativo, adaptado de [10]

3. Análise Qualitativa de Ferramentas de Gerenciamento de Requisitos

Antes da construção da ferramenta considerada adequada para gerenciar requisitos em equipes distribuídas, foi realizado um estudo envolvendo ferramentas de gerenciamento de requisitos comerciais, *free e open source*. O resultado dessa avaliação serve para validar a importância do protótipo desenvolvido, servindo também para a definição dos requisitos da proposta.

Esse estudo foi feito através da técnica *feature analysis*, onde foram identificadas as funcionalidades necessárias para um suporte efetivo à colaboração, com posterior verificação da presença dessas funcionalidades nas ferramentas avaliadas.

3.1. Requisitos para suporte à colaboração

Para realizar a avaliação, primeiramente é necessário identificar os critérios. Para isso, foi definido um conjunto de requisitos desejáveis em ferramentas de gerenciamento de requisitos. Esse levantamento foi feito através de estudos relacionados a ferramentas cooperativas [9] [11] [5] e engenharia de requisitos distribuída [12].

Aspectos de suporte a grupos costumam ser utilizados na literatura como forma de especificar as funcionalidades das ferramentas cooperativas em geral [13]. Por isso, os requisitos identificados foram mapeados para *features* (F1 a F13), as quais foram agrupadas de acordo com estes aspectos [13].

➤ **Comunicação – conectividade e ligação:** O primeiro obstáculo à cooperação é vencer o isolamento e a distância entre cada membro de uma equipe de trabalho, ou seja, estabelecer comunicação e a conectividade entre as partes [13].

F1 – Manutenção de requisitos através de interface web: Deve ser possível a manutenção de requisitos através da *web*, para facilitar o uso pela equipe distribuída. Além disso, a visualização através da *web* permite resolver problemas de interoperabilidade, bem como a validação por clientes remotos.

F2 – Autoria Cooperativa: Deve possibilitar o acesso de múltiplos usuários, na edição (síncrona ou assíncrona) de requisitos, identificando através de um mecanismo adequado os usuários que realizaram modificações.

F3 – Troca de Mensagens: Deve ser possível comunicar com os demais participantes, através de fóruns e mensagens instantâneas, por exemplo. Essa funcionalidade também deve estar presente de forma não vinculada a requisitos, pois nas fases iniciais os requisitos não foram definidos.

F4 – Registro de comentários: Suporte a inserção de comentários, dúvidas, e anotações vinculadas a um determinado requisito.

➤ **Coordenação – acompanhamento e produção:** Esse conceito está relacionado ao acompanhamento de atividades. Especificar como a interação se dará, definir restrições e controlar a execução das tarefas são questões que precisam estar presentes para garantir a produtividade e o sucesso do grupo [13].

F5 – Suporte a execução do processo: A ferramenta deve possibilitar algum tipo de suporte à execução do processo, por exemplo, através de definição de listas de itens a fazer, registro de *templates*, registro e *follow-up* de reuniões.

F6 – Controle de Acesso: Deve permitir a criação de permissões para alteração/visualização de requisitos, de forma individual ou para um grupo ou perfil.

F7 – Baseline e controle de versões: deve ser possível criar *baselines* para controlar requisitos. Além disso, deve ser possível comparar diferentes versões de *baselines*.

F8 – Automação na geração de artefatos: Apesar dessa *feature* apresentar relação com a F5, a geração adequada dos artefatos é imprescindível em uma ferramenta de gerenciamento de requisitos e por isso é tratada como um *feature* diferente.

➤ **Memória de Grupo – registro e histórico de interações:** O suporte a memória de grupo compreende mecanismos que identifiquem as razões por detrás de decisões. Para isso, é necessário capturar as informações e organizá-las de maneira associativa, que englobe não só os produtos gerados (conhecimento formal), como também a experiência e o conhecimento para esta solução (conhecimento informal) [13].

F9 – Base de Conhecimento: A ferramenta deve possuir uma base de conhecimento sobre o projeto, onde estão disponíveis informações sobre o domínio do problema, auxiliando os desenvolvedores na definição e entendimento de requisitos.

F10 – Manutenção de *rationale*: A ferramenta deve apresentar um mecanismo para manutenção do *rationale* de requisitos, por exemplo, através da vinculação com arquivos e discussões de origem.

F11 – Histórico de alteração: Permitir registrar quem, porque, quando e como uma determinada alteração foi realizada ou apenas solicitada. Além disso, alguns desses registros devem ser automáticos.

➤ **Percepção – contexto e localização:** O conceito de percepção busca prover recursos para que todos tenham a noção do contexto de suas atividades dentro do contexto geral do processo, para compreender como o

resultado gerado pelas atividades alheias podem interferir nas suas atividades [13].

F12 – Status do requisito: Consultar o progresso de um requisito (proposto, aceito, entre outros) através de um mecanismo adequado durante o ciclo de vida do projeto.

F13 – Pesquisas *ad hoc*: Consultar requisitos de acordo alguns critérios importantes, tais como: data, responsável, entre outros.

3.2. Quadro comparativo

As ferramentas analisadas estão presentes em uma lista abrangente de ferramentas para gerenciamento de requisitos, construída a partir de um *survey* realizado pelo *International Council On Systems Engineering* [14]. Foi acrescentado a esta lista, ferramentas *open source* que já haviam disponibilizado *releases* nos *sites* pesquisados (*sourceforge* e *tigris*). O protótipo desenvolvido nesse trabalho (CODIPSE-Req) também foi adicionado.

As ferramentas foram classificadas em: (C) – Comercial, (F) – *Free*, e (O) – *Open Source*. Uma ferramenta é considerada *free* se for disponibilizada por uma licença que permite a sua utilização de forma gratuita, mas não necessariamente disponibiliza seu código fonte. Uma ferramenta *open source*, é disponibilizada sob uma licença que torna disponível seu código fonte, entretanto não é necessariamente gratuita. As ferramentas *open source* presentes nessa comparação, são também *free*. Em relação à presença de determinada *feature*, será utilizada a seguinte legenda: (✓) – Quando a funcionalidade estiver presente, (✗) – Quando a funcionalidade não estiver presente e (⊗) – Quando a funcionalidade estiver presente parcialmente.

Tabela 1 - Comparação de Ferramentas de Gerenciamento de Requisitos

No.	Ferramenta	Tipo	Comunicação				Coordenação				Mém. Grupo			Percepção	
			F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
1	Accept Planner	C	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	Analyst Pro	C	✗	✓	✗	✗	⊗	✓	⊗	✓	✗	✓	⊗	✓	✓
3	CaliberRM	C	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✓	✓	✓
4	CARE 3.2	C	⊗	✓	✗	✓	✗	✓	⊗	⊗	✗	⊗	✓	✓	✓
5	Catalyze	C	⊗	✓	✗	✓	✗	✓	✓	✓	✗	⊗	✓	✓	✓
6	CORE 5.1	C	⊗	✓	✗	✓	✗	✓	⊗	✓	✗	⊗	✓	✓	✓
7	Cradle 5.2	C	✗	✓	✗	✓	✗	✗	⊗	✓	✗	⊗	✓	✓	✗
8	Doors	C	✓	✓	✗	✓	⊗	✓	✓	✓	✗	⊗	✓	✓	✓
9	DRES	O	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	⊗	✓	✗
10	Envision VIP	C	✗	✓	✗	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓
11	GatherSpace	F	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	⊗	✗	✗

12	Rational RequisitePro	C	⊗	✓	⊗	✓	✗	⊗	✓	⊗	✗	⊗	✓	✓	✓
13	IRqA 3.0	C	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓	✓
14	Jeremia	O	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
15	Kollabnet Editor	C	✗	✓	✗	✗	✗	⊗	⊗	⊗	⊗	⊗	✓	⊗	✗
16	PACE	C	✓	✓	✓	✓	⊗	✓	⊗	✓	✗	⊗	✓	✓	✓
17	RaQuest 2.1	C	✗	✓	✗	✓	✗	✓	✓	✗	⊗	✗	✓	✓	✓
18	RDT 3.0.2	C	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✗
19	RMTtrak	C	✗	✗	✗	✗	✗	⊗	⊗	✓	✗	✗	✓	✓	✓
20	SoftREQ	F	✓	✗	✗	✗	✗	✗	⊗	⊗	✗	✗	✗	✓	✓
21	SpeeDEV Rel 4.0	C	✓	✓	✓	✓	⊗	✓	✓	✓	✗	⊗	✓	✓	✓
22	TrueReq Product Desktop	C	⊗	✓	⊗	✓	⊗	✓	⊗	✓	✗	⊗	✓	✓	✓
23	XTie-RT	C	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓
24	CODIPSE-Req	O	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗	✓	✓

Apesar do processo de avaliação ter sido bastante detalhado, ele foi baseado principalmente em documentos disponíveis nos sites das ferramentas. Uma análise mais detalhada, utilizando versões de avaliação ainda deve ser realizada.

Foi percebido que a maioria das ferramentas estudadas se preocupa com automação, e não fornece um suporte adequado para a comunicação e colaboração, o que dificulta sua adoção por equipes distribuídas.

Outra constatação interessante é que não existe atualmente uma ferramenta de gerenciamento de requisitos (para desenvolvimento distribuído) *free* ou *open source* que possua as funcionalidades necessárias para o gerenciamento de requisitos em si (com exceção do protótipo desenvolvido neste trabalho, todas não permitem nem ao menos a exportação dos dados cadastrados).

4. A ferramenta CODIPSE-Req

A ferramenta foi construída como parte do projeto CODIPSE (*COoperative and DIstributed Process Support Environment*) (disponível em <http://codipse.tigris.org>). O projeto CODIPSE é um projeto de software livre cujo objetivo é a idealização de um PSEE (*Process-Centered Software Engineering Environment*) que forneça suporte ao trabalho

cooperativo em equipes distribuídas. Dentro do contexto do ambiente CODIPSE, a ferramenta CODIPSE-Req está relacionado à fase de engenharia de requisitos, onde diferentes *stakeholders* podem se comunicar e colaborar.

Nesta primeira versão a ferramenta CODIPSE-Req foi desenvolvida como uma aplicação para o ambiente eGroupware e disponibilizada sob uma licença *open source*, da mesma forma que o ambiente citado. Essa integração permitiu a utilização e interação com as diferentes aplicações já disponibilizadas por este ambiente, facilitando a etapa de implementação.

Dessa forma, a proposta está embasada na utilização de um sistema *open source* existente (eGroupware), onde um novo subsistema (CODIPSE-Req) foi desenvolvido e acoplado.

4.1. Requisitos

A figura 2 apresenta o modelo de casos de uso, onde é possível visualizar os dois subsistemas e a fronteira entre eles. Os casos de uso presentes no subsistema eGroupware já estão disponíveis através do ambiente e representam somente um subconjunto dos casos de uso disponíveis atualmente. Os demais, referentes ao subsistema CODIPSE-Req, foram definidos e implementados nesse trabalho.

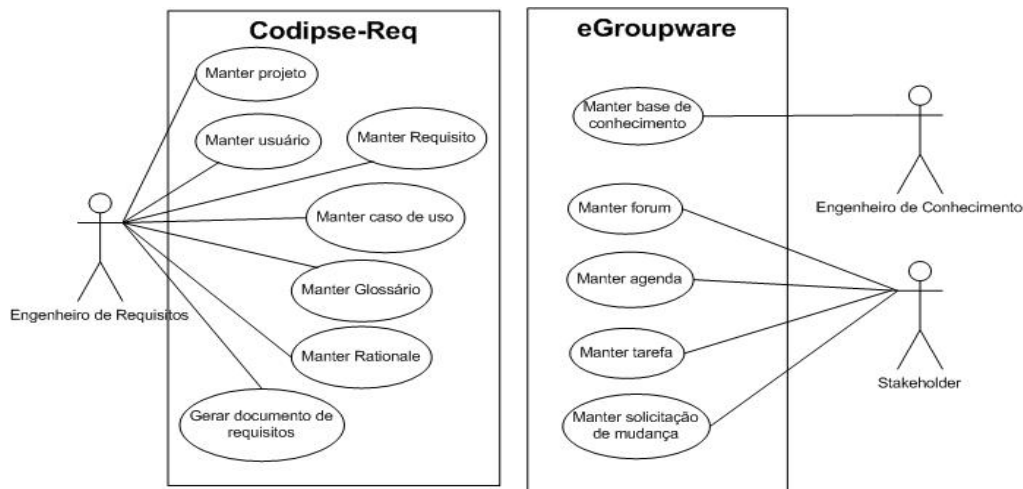


Figura 2 - Diagrama de caso de uso principal

Devido ao fato da proposta utilizar aplicações disponibilizadas pelo ambiente eGroupware, apenas aquelas referentes ao subsistema CODIPSE-Req serão apresentadas em detalhes. A seguir serão descritos em linhas gerais os casos de uso do subsistema CODIPSE-Req.

R1 - Manter Projeto: O sistema possibilita a inclusão, alteração e exclusão de diversos projetos. Para cada projeto, deve-se identificar o seu nome, visão¹, usuários, casos de uso, requisitos e glossário.

R2 - Manter Usuário: O sistema possibilita a inclusão, alteração e exclusão de usuários (ou atores) que participam do sistema.

R3 - Manter Requisito: O sistema possibilita a inclusão, alteração e exclusão de requisitos funcionais e não funcionais.

R4 - Manter Casos de Uso: O sistema possibilita a inclusão, alteração e exclusão de casos de uso, identificando os atores que os realizam, pré-condições, pós-condições, fluxo principal e fluxo alternativo.

R5 - Manter Glossário: O sistema possibilita a inclusão, alteração e exclusão de termos para compor o glossário do sistema.

R6 - Manter Rationale: O sistema deve permitir o registro de rationale (motivo, razão), para que seja possível identificar as decisões que levaram a definição do estado atual da especificação de requisitos. Para isso, o sistema permite a inclusão, alteração e exclusão de links (ligações entre os registros) em nível de registros de projeto, casos de uso ou requisitos, com qualquer informação que tenha originado modificações. Por exemplo, um projeto deve apresentar links para arquivos de documentos que descrevem sistemas relacionados que tenham sido utilizados na especificação e mensagens em fóruns onde se tenha discutido algo sobre o sistema, entre outros.

R7 - Gerar documento de requisitos: O sistema permite a geração de um Documento de Especificação de Requisitos de Software, a partir de um template pré-definido, incluindo todas as informações previamente cadastradas.

4.2. Arquitetura

Durante a fase de análise e projeto foi construído (entre outros artefatos) o modelo de classes e dependência entre pacotes do sistema, apresentado na figura 3. Através do modelo é possível visualizar as entidades, bem como a sua dependência com as demais aplicações do eGroupware. Essas dependências tiveram de ser estabelecidas devido à necessidade na construção de links para manutenção de rationale.

Um Projeto possui Ator, Caso de Uso, Requisito (o atributo tipo identifica se é funcional ou não funcional) e Glossário (composto de termos seguindo a definição de LAL²).

¹ O objetivo de descrever a visão do sistema é expor as necessidades e funcionalidades gerais do sistema, identificados a partir de coleta inicial de requisitos.

² O principal objetivo do LAL (Léxico Ampliado da Linguagem) é registrar a linguagem utilizada pelos atores do UdI (Universo de Informações). O LAL do UdI é composto por entradas, onde cada entrada está associada a um símbolo (palavra ou frase) da linguagem do UdI. Cada símbolo pode possuir sinônimos e é descrito através de noções e impactos. As noções descrevem o significado e as relações fundamentais de existência do símbolo com outros símbolos (denotação). Os impactos descrevem os efeitos causados pelo uso, ou ocorrência, do símbolo no UdI. Dependendo do símbolo que descrevem, as entradas podem ser classificadas como sujeito, verbo, objeto e estado (predicativo) [15].

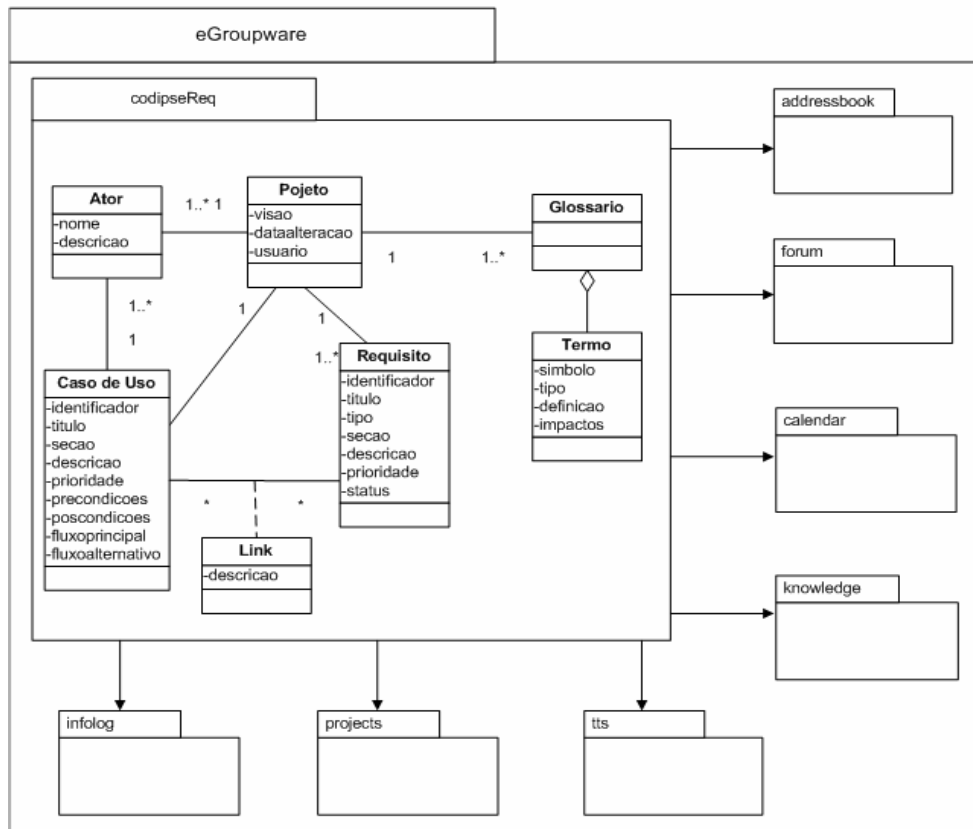


Figura 3 - Diagrama de classes e dependência de pacotes do módulo "Gerenciamento de Requisitos"

Os pacotes utilizados foram: addressbook (disponibiliza a lista de contatos compartilhado), forum (aplicação de fórum), calendar (disponibiliza a aplicação de agenda), knowledge (base de conhecimento), tts (*Trouble Ticket System*, um *Issue Tracker* semelhante ao Bugzilla [16]), projects (onde é possível acompanhar o custo de um projeto), infolog (para registro e acompanhamento de tarefas).

A aplicação desenvolvida (CODIPSE-Req) possui três módulos principais:

(1) Gerenciamento de Requisitos: agrupando funcionalidades para manutenção de projeto como visão/escopo, casos de uso, requisitos funcionais e não funcionais, glossário e atores;

(2) Manutenção do Rationale: agrupa funcionalidade para criar *links* com as ferramentas de colaboração do eGroupware. Os *links* podem ser criados em nível de projeto, casos de uso ou requisitos, permitindo a completa manutenção do *rationale* de requisitos;

(3) Exportação: esse módulo é responsável pela geração intermediária (em XML) dos dados presentes em um determinado projeto. A partir dessa versão e através de transformação XSL é possível exportar as informações para qualquer formato. O único formato suportado atualmente é o Microsoft Word 2003.

A aplicação foi implementada utilizando a linguagem de programação PHP, orientação a objetos e arquitetura em três camadas, os quais também são adotados pelo sistema eGroupware. As telas da aplicação serão mostradas no estudo caso da seção a seguir.

5. Estudo de Caso

O objetivo deste estudo de caso é apresentar um exemplo de uso do protótipo CODIPSE-Req e dessa forma demonstrar a aplicabilidade e benefícios da proposta. O processo de Engenharia de Requisitos é essencialmente colaborativo e depende da qualidade das interações realizadas, além da experiência dos envolvidos. Se essas interações forem realizadas de forma distribuída, o ambiente deve apresentar diferentes formas de comunicação.

O estudo de caso irá apresentar algumas telas da aplicação seguindo o processo de engenheiro de requisitos definido em [6], o qual define as seguintes atividades: Elicitação, Análise e Negociação, Documentação e Validação. Ainda, será mostrada a atividade de rastreamento, necessária para manutenção de *rationale*.

Durante a atividade de "Elicitação de Requisitos", são marcadas reuniões na agenda de eventos (aplicação disponibilizada pelo eGroupware). Após a realização das mesmas, o *log* da ferramenta *MSN Messenger* é anexado e são inseridos registros sobre o que foi decidido ou discutido. O Engenheiro de Requisitos então, cadastra os rascunhos iniciais dos requisitos e termos utilizados na ferramenta CODIPSE-Req. As atas de reuniões correspondem às anotações feitas na agenda de eventos e ao *log* (conforme figura 4).

Agenda de Eventos - Exibir

Título: Reuniao de Elicitacao
Descrição: Discussao sobre requisitos
Localização: Messenger
Projeto: XPlanner [XPlanner]
Início Data/hora: 2005/11/15 - 13:00
Data/Hora do término: 2005/11/15 - 14:00
Prioridade: Normal
Acesso: público
Participantes: [jacmorel] Jacques Morel (Não Responde)
[mregazzi] Matteo Regazzi (Não Responde)
[osandum] Ole Sandum (Não Responde)
[charlieevett] Charlie Evett (Não Responde)
[raquel] Raquel (Aceito)
Criado por: [raquel] Raquel
Atualizado: 2005/12/21 - 17:42

Tarefas Adicionar: Pendências Chamadas Telefônicas Nota

Tipo ...	Assunto	Data Inicial	Dono	Última alteração	Sub	Ação
Status ...	Descrição	Data Final	Responsável			
Nota Encerrado	Ata da Reunião Decisoes Firmadas: 1) Decisao 1 2) Decisao 2 reuniao.log	2005/12/21 2005/12/21	[raquel] Raquel	2005/12/21 17:51 [raquel] Raquel	adicionar uma Sub	Editar Remover

Mês: Novembro 2005 **Ano:** 2005

Figura 4 - Elicitação de Requisitos

Em paralelo à elicitação de requisitos, ocorre a Análise e Negociação de Requisitos. Nessa atividade, são realizadas reuniões para discutir dúvidas e inconsistências. Para manter o *rationale*, as alterações

são mantidas através de *links* com qualquer interação que tenha levado à sua alteração (conforme mostrado na figura 5).

codipse

[raquel] Raquel - Terça 2005/12/20

Especificação de Requisitos - Editar dicas de formatacao**

Projeto: XPlanner

Criar novos links

Localizar:
arquivo a:

Links exist:
Projetos: Sistema de Tickets de Processos (STP)
Requisitos: word 157k
Arquivo: 5k

Responsável: [raquel] Raquel Última alteração: 2005/11/15 05:18

Figura 5 - Manutenção de Rationale de Requisitos

A “Documentação de Requisitos” vem ocorrendo desde a etapa de elicitação, com a versão rascunho dos requisitos cadastrados no sistema. A figura 6 apresenta a tela para cadastro de caso de uso, onde é possível

inserir descrição, atores, pré-condições, pós-condições, fluxo principal, fluxo alternativo e links para manutenção de *rationale*.

Figura 6 - Manutenção de Caso de uso

No entanto, durante a etapa de documentação, os requisitos se transformam em versões finais, prontos para serem implementados após validação. Através do protótipo é possível exportar as informações

cadastradas para o Microsoft Word 2003, conforme figura 7.

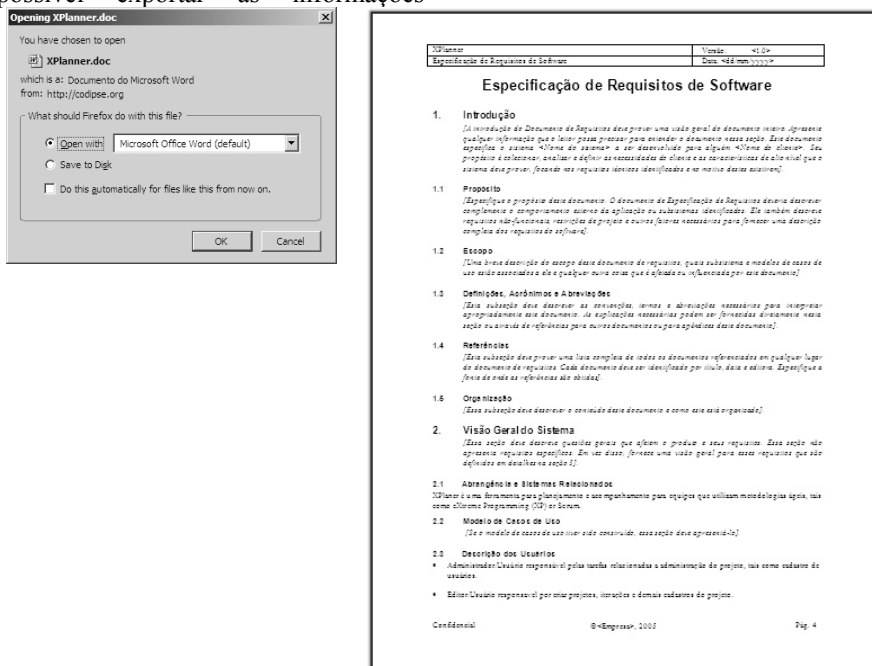


Figura 7 - Geração do documento de especificação de requisitos

A “Validação de Requisitos” ocorre através de reuniões síncronas (onde novamente o log das discussões deve ser mantido), ou através do aceite dos

responsáveis. Em qualquer caso, o status do requisito deve ser alterado para “aceito” e informações sobre a validação devem ser inseridas no controle de *rationale*.

6. Considerações Finais e Trabalhos Futuros

A utilização de tecnologias de colaboração tem sido apontada como um dos principais fatores para auxiliar a engenharia de requisitos distribuída [17] [7] [8].

A *feature analysis* das ferramentas de gerenciamento de requisitos constatou que a maioria das ferramentas de gerenciamento de requisitos não fornece suporte adequado às equipes distribuídas e identificou a inexistência de ferramentas *open source* ou *free* que forneça a automação necessária para um gerenciamento eficiente.

Esse artigo apresentou um protótipo para gerenciamento de requisitos construído para auxiliar a verificação da viabilidade da proposta feita neste trabalho. A aplicação utilizou um *groupware open source*, o *eGroupware*, sendo acoplado a este. O protótipo desenvolvido foi disponibilizado como software livre [18] e é parte da proposta da plataforma CODIPSE, ainda em desenvolvimento.

A principal dificuldade encontrada esteve relacionada ao entendimento da arquitetura do ambiente *eGroupware* e da linguagem de programação PHP. Além disso, a falta de documentação dificultou a tarefa de integração da aplicação desenvolvida com as demais aplicações do ambiente. Depois de solucionados estes problemas, o desenvolvimento do protótipo ocorreu de forma natural e o ambiente pôde ser completamente implementado.

Um estudo de caso foi apresentado para demonstração dos benefícios da proposta. No entanto, ainda é necessária a validação em um número maior de estudos de caso e/ou realização de experimentos, não realizados devido à falta de tempo hábil.

7. Referências

- [1] J. D. Herbsleb and D. Moitra, "Global Software Development," IEEE Software, 2001.
- [2] R. Prikładnicki, "MuNDDoS Um modelo de Referência para o Desenvolvimento Distribuído de Software." Dissertação de mestrado da PUCRS, Porto Alegre, 2003.
- [3] L. T. Lopes, "Um Modelo de Processo de Engenharia de Requisitos para Ambientes de Desenvolvimento Distribuído de Software." Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática, 2004.
- [4] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," 2000.
- [5] D. E. Herlea and S. Greenberg, "Using a Groupware Space for Distributed Requirements Engineering," 1998.
- [6] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques* John Wiley & Sons Ltda, ISBN: 0-471-97208-8, 1998.
- [7] D. Damian and D. Zowghi, "Requirements Engineering challenges in multi-site software development organizations," 2003.
- [8] R. Prikładnicki, J. L. N. Audy, and R. Evaristo, "Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context," 2003.
- [9] D. Togneri, R. Falbo, and C. Menezes, "Supporting Cooperative Requirements Engineering with an Automated Tool," Anais do V WER, 2002.
- [10] V. Ambriola and V. Gervasi, "The Case for Cooperative Requirement Writing," 1998.
- [11] M. Machado, F. Santos, C. M. L. Werner, and M. R. S. Borges, "Uma infra-estrutura de apoio à aquisição cooperativa de conhecimento em engenharia de domínio," 1999.
- [12] D. Damian and D. Zowghi, "The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization," 2002.
- [13] R. M. Araújo, "Ampliando a cultura de processos de software – um enfoque baseado em groupware e workflow." Tese de doutorado UFRJ - COOPE, 2000.
- [14] INCOSE Tools Database Working Group (DBWG), "Requirements Management Tools Survey," 2005.
- [15] A. C. V. B. Didier, "WRE-Process: Um Processo de Engenharia de Requisitos Baseado no RUP." Dissertação de Mestrado da Universidade Federal de Pernambuco, Centro de Informática, 2003.
- [16] Bugzilla, "Bugzilla," 2006.
- [17] W. J. Lloyd, M. B. Rosson, and J. D. Arthur, "Effectiveness of Elicitation Techniques in Distributed Requirements Engineering," 2002.
- [18] CODIPSE, "COoperative and DIstributed Process Support Environment," 2005.