

Suporte a Engenharia de Requisitos com Equipes Distribuídas

Regiane Andrade Brito, Alexandre Marcos Lins de Vasconcelos

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851, 50732-970, Recife – PE – Brazil

{rab3, amlv}@cin.ufpe.br

***Abstract.** This paper presents a prototype of a cooperative tool built as an Eclipse plugin to support requirements engineering. This tool support work in distributed teams, providing collaboration and communication among stakeholders. Furthermore, it was developed using a new Eclipse's Project Framework named ECF (Eclipse Communications Framework) that aims to make it easy collaborative applications development.*

***Resumo.** Este artigo apresenta o protótipo de uma ferramenta cooperativa para engenharia de requisitos, construída na forma de um plugin para o Eclipse. A ferramenta auxilia o trabalho de equipes distribuídas, permitindo edição colaborativa e comunicação entre os stakeholders. Além disso, ela foi construída com base em um novo framework do projeto Eclipse, o ECF (Eclipse Communications Framework), que pretende auxiliar a construção de aplicações colaborativas.*

1. Introdução

O desenvolvimento de software com equipes geograficamente distribuídas é uma realidade e várias empresas já utilizam esse estilo de desenvolvimento (Battin et al, 2001) (Prikladnicki et al, 2003). Um dos precursores desse estilo é o desenvolvimento de software livre, onde um grupo de usuários participa de forma colaborativa na construção de um produto que pode ser estudado e modificado livremente por todos os interessados.

Existem alguns fatores que impulsionaram o desenvolvimento distribuído entre grandes empresas de software, dentre eles: necessidade de utilizar recurso capacitado, onde quer que ele esteja; vantagens de estar próximo de diferentes mercados, conhecendo consumidores e condições locais; formação rápida de organizações virtuais e equipes virtuais para explorar as oportunidades do mercado; necessidade de diminuir o tempo de entrega do produto para o mercado (*time to market*), conseguido através de fusos horários diferentes (produção 24 horas por dia) (Herbsleb e Moitra, 2001).

No entanto, estudos (Carmel, 1999) (Herbsleb e Grinter, 1999) indicam que equipes distribuídas enfrentam problemas como: falta de coordenação, comunicação ineficiente, diferenças de fusos horários, diferenças culturais, perda do espírito de equipe e falta de coordenação entre os participantes. Em especial, a disciplina de engenharia de requisitos merece uma maior atenção (Lopes e Audy, 2003), por ser considerada como a maior responsável pela qualidade do sistema final (Nuseibeh e

Easterbrook, 2000) e por ser uma atividade complexa pois requer muita interação entre pessoas com conhecimentos e culturas diferentes.

O suporte adequado de um ambiente de desenvolvimento de software pode auxiliar o trabalho em equipes dispersas. Para isso, esse ambiente deve lidar com problemas como: manter a comunicação informal entre os participantes (Herbsleb e Moitra, 2001) para construção do espírito de equipe (Carmel, 1999) e consciência de trabalho coletivo, coordenação das atividades dos integrantes, percepção do que está sendo produzido por todas as partes envolvidas, gerência de configuração, utilização de um processo padrão ou padronização das atividades e nomenclatura dos artefatos (Battin et al, 2001).

Esse trabalho apresenta um protótipo de uma ferramenta para auxiliar a engenharia de requisitos em equipes distribuídas. Ela permite que colaboradores editem de forma síncrona um documento de requisitos e troquem mensagens instantâneas durante as interações. Esse protótipo foi construído como parte do projeto CODIPSE (codipse.tigris.org), o qual visa a construção de um ADS (ambiente de desenvolvimento de software) para equipes distribuídas.

A seção 2 apresenta uma visão geral sobre engenharia de requisitos e seu contexto em Desenvolvimento Distribuído de Software. A seção 3 trata sobre o suporte a engenharia de requisitos nesse contexto de distribuição. A seção 4 apresenta uma visão geral do ambiente CODIPSE. A seção 5 apresenta o protótipo da ferramenta Codipse-Req. Por fim, a seção 6 apresenta as conclusões, trabalhos futuros e como será a validação do trabalho.

2. Engenharia de Requisitos em DDS

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 uma implementação futura (Nuseibeh e Easterbrook, 2000).

No desenvolvimento com equipes co-localizadas, a coleta de requisitos ocorre em reuniões face a face, considerada a forma mais efetiva de comunicação (Damian et al, 2000). Esses encontros tornam mais fácil também as atividades de negociação e resolução de conflitos.

Em DDS essa atividade tende a ser dificultada por problemas inerentes da distribuição geográfica, além dos problemas já conhecidos da área como dificuldade de comunicação entre os *stakeholders* (Kotonya e Sommerville, 1998). Alguns estudos de caso relatam experiências na área de engenharia de requisitos distribuída, apresentando as lições aprendidas (Damian e Zowghi, 2003) (Prikladnicki et al, 2003).

3. Suporte a Engenharia de Requisitos Distribuída

Fornecer suporte a Engenharia de Requisitos Distribuída envolve tanto definir um processo adequado quanto uma ferramenta colaborativa para auxiliar a execução desse processo. Essa ferramenta deve ser definida através da integração com a área de CSCW (*Computer Supported Cooperative Work*) para que a mesma possua funcionalidades como: comunicação, coordenação, percepção e memória de grupo (Araújo, 2000).

Em relação ao processo de engenharia de requisitos, a figura 1 apresenta como este deve ser caracterizado, em relação ao processo tradicional. No site da ferramenta (codipse.tigris.org) é possível fazer o *download* de um processo definido (baseado no SPEM). A apresentação desse processo não está no escopo desse trabalho.

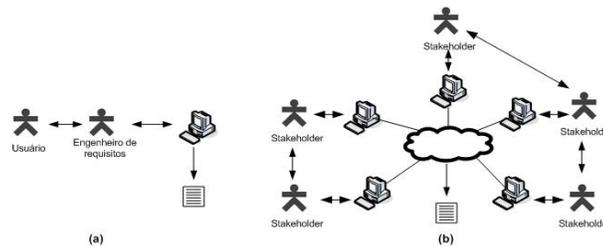


Figura 1: (a) um processo mediado por um desenvolvedor (b) um processo totalmente cooperativo

A idéia de construir um ambiente colaborativo para Engenharia de Requisitos não é nova. Existem algumas propostas na literatura, como o ambiente CRETA (Togneri et al, 2002) que é baseado na web e permite a integração com um processo de engenharia de requisitos e o Teamwave instanciado para engenharia de requisitos (Herlea e Greenberg, 1998). No entanto, essas ferramentas não estão integradas a um ambiente de desenvolvimento de software, e não há indícios de que seu desenvolvimento esteja sendo continuado.

4. A proposta CODIPSE (COoperative and DIstributed Process Support Environemnt)

O ambiente CODIPSE está sendo desenvolvido como um projeto de software livre e sua principal motivação é fornecer uma infra-estrutura ao mesmo tempo informal e formal aos desenvolvedores, facilitando também com que novos integrantes entendam facilmente tudo o que já foi construído.

Para isso, o ambiente visa integrar características de PSEE's (*Process-Centered Software Engineering Environemnts*), metodologias ágeis e software livre, fornecendo um ambiente descentralizado e colaborativo para desenvolvimento de software.

No que diz respeito à PSEE's, as características incorporadas destes são aquelas relacionadas com a definição e execução de processos de software, bem como o suporte a colaboração. Além disso, o ambiente não visa obrigar a execução de um processo e sim fornecer orientação ativa para os desenvolvedores do ambiente (Downson, 1993).

Em relação às metodologias ágeis, a forte ênfase em comunicação inspirou o fornecimento de um rico conjunto de formas de interação, embasada pelos estudos realizados em *groupware*. Além disso, através das ferramentas de comunicação pode-se estar em constante contato com o cliente e embora o mesmo não esteja *on site*, ele pode participar de reuniões da mesma forma que os participantes distantes geograficamente.

A relação com software livre, busca estudar a forma como esse estilo de desenvolvimento vem sendo realizado, através de sistemas baseados inteiramente na internet, com ferramentas para coordenação e comunicação dos participantes e utilização de sistemas de controle de versão, controle de itens a fazer e *bugs*, além de e-

mail e fóruns de discussão. Além disso, foi o estilo de desenvolvimento distribuído que serviu como experiência para definição desse trabalho (Brito et al, 2004)

Aspectos de suporte a grupos costumam ser utilizados na literatura como forma de especificar as funcionalidades das ferramentas cooperativas em geral (Araújo, 2000). A figura 2 apresenta o sistema em relação aos elementos de CSCW identificados.

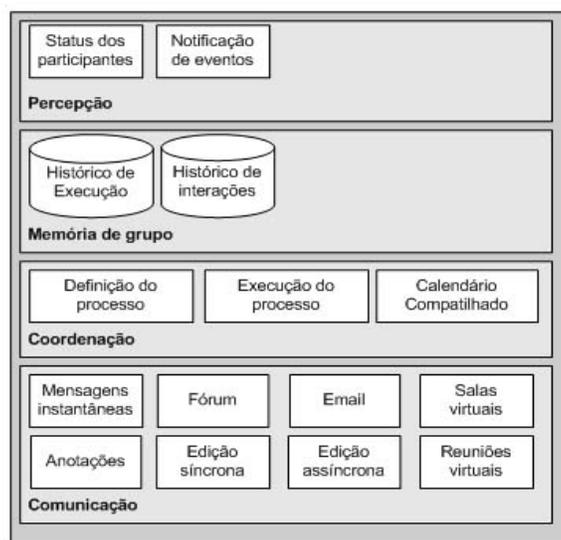


Figura 2 – Serviços de colaboração do CODIPSE

Um mapeamento entre os aspectos de suporte a grupos (Araújo, 2000) e os serviços do sistema, será apresentado a seguir.

(a) Comunicação – conectividade e ligação: O primeiro obstáculo para a 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 envolvidas (Araújo, 2000). Portanto, a comunicação é o aspecto mais importante no ambiente e é coberta de diferentes formas. Estão previstos fóruns, e-mail, ferramentas para edição assíncrona, serviços de anotação para revisão de artefatos, mensagens instantâneas, edição síncrona, reuniões virtuais e salas virtuais, onde é possível visualizar os demais participantes.

(b) Coordenação – acompanhamento e produção: O aspecto fundamental em relação à coordenação, diz respeito ao acompanhamento das atividades. Especificar como a interação se dará, definir restrições e controlar a execução das tarefas são questões que precisam ser suportadas para garantir a produtividade e o sucesso dos objetivos do grupo (Araújo, 2000). Os serviços de coordenação oferecidos estão diretamente ligados ao processo de software definido e em execução, onde é possível visualizar o processo e acompanhar sua execução através de listas de itens a fazer. Além disso, um calendário compartilhado pode ser acessado e atualizado pelos participantes.

(c) Memória de Grupo – registro e histórico de interações: O suporte a memória de grupo compreende mecanismos que identifiquem claramente as razões por detrás de cada decisão. 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)

(Araújo, 2000). Para isso, serão armazenados definições e histórico de execução do processo (geração de artefatos, definição da equipe, entre outros), bem como as interações realizadas entre os participantes para esclarecimento de dúvidas ou tomada de decisões. Todas essas informações fornecem *feedback* para o processo e ajuda novos desenvolvedores a entender o contexto no qual uma determinada decisão foi tomada.

(d) 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 (Araújo, 2000). Em relação à percepção, o sistema realiza a adequação da interface do usuário, para que estejam visíveis notificações e itens importantes no contexto de trabalho do usuário. Além disso, é possível perceber o status (*online*, *offline*, entre outros) dos participantes em um determinado momento.

Muitos dos serviços apresentados são encontrados em *groupwares* existentes. Embora um *groupware* possa ser utilizado de forma isolada, a conexão com um ADS possibilita mediar as interações e vincular estas à execução do processo.

5. O protótipo Codipse-Req

Existem poucas propostas na literatura para suporte a engenharia de requisitos feita de forma colaborativa. A maioria dos ambientes tratam da colaboração durante a edição de modelos de análise e projeto e código fonte (Grundy et al, 2000).

A ferramenta foi construída como parte do projeto CODIPSE (*COoperative and DIstributed Process Suport Environment*) e é disponibilizada pela licença CPL (<http://www.eclipse.org/legal/cpl-v10.html>). O *download* pode ser feito no site do projeto no Tigris (codipse.tigris.org), através de “Documents & Files”.

O Projeto Eclipse disponibiliza o EMF (*Eclipse Modelling Framework*). Este *framework* possibilita a integração entre modelos e código fonte, possibilitando a transição de um para o outro. Através de modelos como um XML Schema ou UML, é possível gerar automaticamente as classes e interfaces correspondentes, bem como editores (usados para criar instâncias do modelo).

Recentemente, o Eclipse iniciou o desenvolvimento de um novo *framework* denominado ECF (*Eclipse Communications Framework*). Esse projeto pretende facilitar a criação de aplicações colaborativas para a plataforma Eclipse, fornecendo uma API para mensagens síncronas e assíncronas, as quais podem ser enviadas entre pessoas, entre *plugins* ou entre os dois.

A figura 2 apresenta a arquitetura do ambiente, utilizando extensões dos *frameworks* citados acima. O Modelo SDO (*Service Data Objects*) é gerado utilizando o EMF, a partir de um XML Schema para representação de um documento de requisitos. O SDO é um padrão para representação de dados, o qual possui implementação no EMF. Esse padrão foi usado porque o ECF possui suporte a ele. Um editor compartilhado foi construído para que uma instância desse modelo pudesse ser criada de forma síncrona por um engenheiro de requisitos e editado pelos demais *stakeholders*.

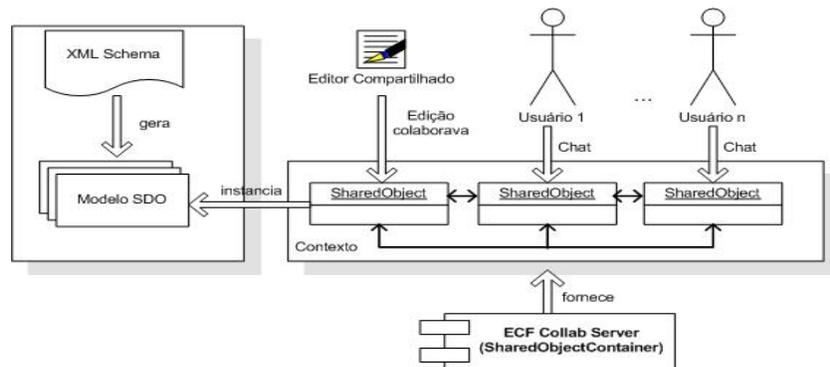


Figura 3 – Arquitetura da ferramenta

A aplicação de *chat* foi construída com base em um exemplo disponibilizado pelo ECF e adaptações foram feitas para que este incluísse o papel do participante no processo. Os conceitos principais no ECF são: *SharedObject* e *SharedObjectContainer*. *SharedObject* representa objetos compartilhados que podem comunicar-se entre si (o modelo e os usuários são representados dessa forma). O *SharedObjectContainer* define um contexto (um projeto, por exemplo) para que esses objetos possam trocar mensagens. Além disso, ele fornece o protocolo de comunicação que deve ser utilizado (o ECF Collab Server utiliza TCP).

O *plugin* possui as funcionalidades de edição síncrona de um modelo, onde é possível registrar requisitos funcionais, não funcionais, casos de uso e cenários. Além disso, é possível realizar *chat* e enviar arquivos. Na tela de login, é possível o usuário escolher o seu papel no processo, facilitando assim a identificação pelos demais participantes, além da possibilidade de registrá-lo posteriormente no *log*.

Após isso, os participantes podem editar um modelo pré-existente como o mostrado na figura 4. Nesse exemplo, foram criadas seções para requisitos não-funcionais, casos de uso e requisitos funcionais. No entanto, é possível construir um documento de especificação de requisitos completo através desse modelo. Sempre que um usuário salva o modelo, as alterações são propagadas para os demais usuários que estão editando.



Figura 4 – Edição colaborativa

Além disso, durante essa sessão, os participantes podem trocar mensagens instantâneas, bem como visualizar o papel dos demais envolvidos, como mostrado na figura 5. Os *logs* dessas conversas podem então ser vinculados às alterações que foram

realizadas no modelo referente, permitindo compreender o contexto no qual decisões foram tomadas, sendo crucial para entender o *rationale* dos requisitos.

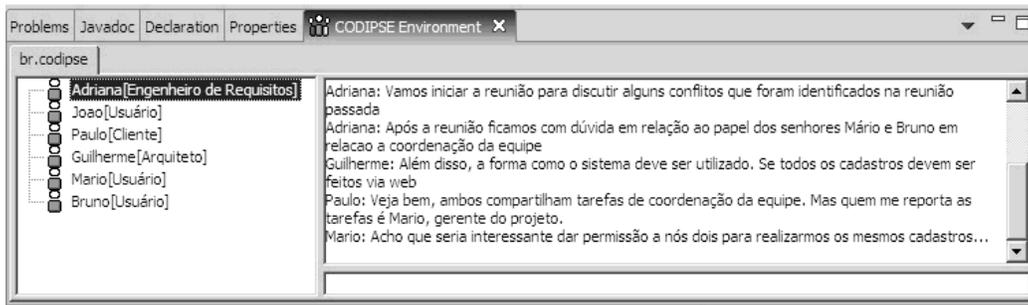


Figura 5 – Chat entre os participantes

A maior dificuldade na implementação foi o entendimento do *framework* ECF e da arquitetura de *plugins* do eclipse. No entanto, a quantidade de código criado foi relativamente pequena, visto a reutilização do código disponibilizado como exemplo pelo ECF e pela utilização do XML Schema já definido. Após isso, o trabalho foi concentrado na criação do editor do modelo, pois houve a adaptação para que o mesmo pudesse ser feito de forma colaborativa e utilizando o padrão SDO.

6. Considerações finais e Trabalhos Futuros

Esse artigo apresentou a construção de uma ferramenta cooperativa para engenharia de requisitos, através da utilização dos *frameworks* ECF e EMF disponibilizados pelo projeto Eclipse. A utilização de tecnologias de colaboração tem sido apontada como um dos principais fatores para auxiliar a engenharia de requisitos distribuída (Lloyd et al, 2002) (Damian e Zowghi, 2003) (Prikladnicki et al, 2003).

O Eclipse é uma plataforma para integração de qualquer tipo de aplicação, através de uma arquitetura de *plugins* robusta e bem definida. Além disso, é uma IDE bastante popular, principalmente na comunidade *open source*. Dessa forma, a construção de uma ferramenta como a proposta neste trabalho visa também incentivar as atividades de engenharia de requisitos em comunidades de software livre.

Embora a ferramenta apresentada ainda não possua todas as funcionalidades desejáveis para um contexto no qual ela se propõe, os resultados obtidos e as análises dos *frameworks* disponibilizados pelo Eclipse indicam que é possível a fácil construção de soluções mais elaboradas.

Como trabalho futuro está prevista a integração de gerência de conhecimento baseada no processo de software definido para equipes distribuídas disponibilizado no site da ferramenta. O ambiente deve prover suporte para a geração de todos os artefatos e interações previstos nesse processo e incluir funcionalidades para coordenação e memória de grupo. Após a conclusão do ambiente, experimentos serão realizados dentro de uma disciplina de graduação, onde as equipes utilizarão a ferramenta e a qualidade dos artefatos gerados será analisada.

7. Referências

- ARAÚJO, R. M. "Ampliando a cultura de processos de software – um enfoque baseado em groupware e workflow". Tese de doutorado. UFRJ - COOPE, 2000
- BATTIN, R. D., CROCKER, R., e KREIDLER, J. "Leveraging resource in Global Software Development". IEE Software, 2001
- BRITO, R., FERREIRA, P., SILVA, K. et al. "Uma experiência na implantação de processo em uma fábrica de software livre". VI Simpósio Internacional de Melhoria de Processo de Software, 2004
- CARMEL, E. "Global Software Teams – Collaborating Across Borders and Time Zones". EUA: Prentice Hall, 1999
- DAMIAN, D., ALBERIN, A., e SHAW, M. L. G. "Using Different Communication Media in Requirements Negotiation". IEEE Software, 2000
- DAMIAN, D. e ZOWGHI, D. "Requirements Engineering challenges in multi-site software development organizations". Requirements Engineering Journal, 8, pp.149-160, 2003
- DOWNSON, M. "Software Process Themes and Issues". IEEE Software, 1993
- GRUNDY, J., MUGRIDGE, W., e HOSKING, J. "Constructing Component-Based Software Engineering Environments: Issues and Experiences". Journal of Information and Software Technology, Vol 42, No.2, pp 117-128, 2000
- HERBSLEB, J. D. e GRINTER, R. E. "Splitting the Organization and Integrating the Code: Conway's Law Revisited". Proceedings of ICSE. Los Angeles: IEEE, 1999
- HERBSLEB, J. D. e MOITRA, D. "Global Software Development". IEEE Software, 2001
- HERLEA, D. E. e GREENBERG, S. "Using a Groupware Space for Distributed Requirements Engineering". Proceedings of WET ICE, California, USA, 1998
- KOTONYA, G. e SOMMERVILLE, I. "Requirements Engineering: Processes and Techniques". John Wiley & Sons Ltda, ISBN: 0-471-97208-8, 1998
- LLOYD, W. J., ROSSON, M. B., e ARTHUR, J. D. "Effectiveness of Elicitation Techniques in Distributed Requirements Engineering". ICRE Proceedings, 2002
- LOPES, L. e AUDY, J. L. N. "Em Busca de um Modelo de Referência para Engenharia de Requisitos em Ambientes de Desenvolvimento Distribuído". Anais do VI WER, 2003
- NUSEIBEH, B. e EASTERBROOK, S. "Requirements Engineering: A Roadmap". The Future of Software Engineering, Anthony Finkelstein, pp.37-46, ACM Press, 2000
- PRIKLADNICKI, R., AUDY, J. L. N., e EVARISTO, R. "Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context". International Workshop on Global Software Development, 2003
- TOGNERI, D., FALBO, R., e MENEZES, C. "Supporting Cooperative Requirements Engineering with an Automated Tool". Anais do V WER, 2002