

Estudo das Interações Humanas no Processo de Desenvolvimento de Software

Cirano Iochpe & Elza Marisa Paiva de Figueiredo

CPGCC-UFRGS

Caixa Postal 15064

91501-970 Porto Alegre - RS

Brasil

e-mail: {elzaiochpe}@inf.ufrgs.br

Abstract: This paper presents the instrumentalization as well as main results of a field study carried out with four software development teams working in three different software companies that are located in South Brazil. Relying on the study, we tried to both get acquainted of how software development really occurs in brazilian companies and to identify the human interactions that take place within the development process and which influence they have on it. As a second step, we analyzed a number of existing software tools in order to find out how they cope with the requirements we inferred from the field study.

Resumo: Este artigo apresenta a instrumentalização e os resultados mais importantes de um trabalho de campo feito com quatro equipes de desenvolvimento de software de três empresas da região da Grande Porto Alegre, Rio Grande do Sul. Os dois maiores objetivos do estudo eram conhecer melhor o processo de desenvolvimento de software que as empresas realmente desenvolvem e identificar quais as interações humanas que ocorrem no processo e como elas influenciam a produtividade e qualidade do processo e do produto final. Com base nos resultados do trabalho, investigou-se como as ferramentas existentes que apoiam o desenvolvimento de software suprem os requisitos identificados.

1. Introdução

O processo de desenvolvimento de software é uma atividade que envolve a participação de vários profissionais, desempenhando diversos papéis durante as várias etapas do processo. Além dos aspectos técnicos, o processo de desenvolvimento de software é reconhecido como uma atividade social composta de interações entre os membros da equipe. Estes membros trabalham de forma cooperativa e interagem para o desenvolvimento de sistemas complexos. Tais interações ocorrem nas diferentes etapas do processo e são de grande importância para a qualidade do produto final.

Como o número de interações que ocorrem entre os profissionais da equipe é grande, elas tomam-se um ponto chave para o sucesso ou fracasso do esforço de desenvolvimento. Assim, decidimos investigar o processo de desenvolvimento de software em detalhes para melhor entender as interações humanas que nele ocorrem. Decidiu-se, então, organizar um estudo de campo, para observar o processo de desenvolvimento de software em vários grupos de diferentes companhias, onde quatro equipes de desenvolvimento de software, localizadas em três empresas distintas, foram observadas durante quatro meses.

O método utilizado pelos observadores durante o estudo é apresentado em [Thio87]. Este método, conhecido por pesquisa-ação, tem como principal objetivo capacitar os pesquisadores a responder com maior eficiência aos problemas da situação em que vivem, sob forma de *diretrizes de ação transformadora*. A pesquisa-ação é um tipo de pesquisa social com base empírica que é concebida e realizada em associação com uma ação ou com a resolução de um problema coletivo. Nela, pesquisadores e participantes da situação ou do problema estão envolvidos de modo cooperativo e participativo.

As observações concentraram-se em três aspectos do processo: no fluxo de atividades do projeto (workflow), nas interações humanas que nele ocorrem; e na influência das últimas na evolução do fluxo de atividades. Em relação às interações, os observadores levantaram seus aspectos operacionais e psicológicos.

Durante o trabalho de campo, percebeu-se que as interações observadas nas quatro equipes de desenvolvimento poderiam ser classificadas. Inicialmente, tentou-se aplicar a classificação apresentada em [Dürr93], o que, mais tarde, constatou-se não ser possível. Partiu-se, então para a uma nova proposta de classificação de interações. Com isso, criou-se uma linguagem de modelagem do workflow que integra a modelagem de interações humanas. Tal linguagem permite a descrição de atividades e interações, combinando símbolos gráficos e texto.

Este artigo esboça como foi realizado o estudo de campo e apresenta os principais resultados através dele obtidos. A partir de tais resultados, infere-se um conjunto de requisitos básicos que devem ser satisfeitos por sistemas de apoio ao desenvolvimento de software. Nossa hipótese é de que tal suporte deve contribuir, de forma decisiva, para o aumento da produtividade do processo e da qualidade do produto final.

O restante deste artigo está organizado como segue. No segundo capítulo, o estudo de campo é apresentado. O Capítulo 3 discorre sobre os resultados da observação realizada. No Capítulo 4, o conjunto de requisitos básicos é discutido. A seguir, no Capítulo 5, algumas ferramentas já existentes são analisadas com base nos requisitos identificados. Por último, algumas conclusões e perspectivas de trabalho futuro são apresentadas no Capítulo 6.

2. Descrição do Processo de Desenvolvimento de Software

A pesquisa de campo foi organizada, seguindo alguns critérios de classificação. O primeiro critério foi classificar o tamanho das equipes de desenvolvimento, tendo como base o número de pessoas que delas participam, incluindo usuários, analistas de negócios, analistas de sistemas, projetistas de sistemas e programadores.

De acordo com nossa classificação, as equipes poderiam enquadrar-se dentro de três

categorias principais: pequena (três pessoas ou menos), média (quatro a seis profissionais) e grande (mais de seis pessoas). Foram escolhidas, para participarem do estudo, duas equipes de desenvolvimento médias e duas grandes, de três empresas distintas.

Um segundo critério de classificação usado foi quanto aos tipos de ambiente de desenvolvimento de software. Como descrito em [Fig95]¹, vários tipos de ambiente foram identificados, de acordo com o vínculo da equipe de desenvolvimento com a empresa usuária e o público alvo do software. Destacam-se os tipos:

- Dedicado, onde o desenvolvimento de sistemas específicos ocorre dentro da própria empresa;
- Autônomo, onde a produção é feita para resolver um problema comum a um determinado nicho de mercado;
- Encomenda, onde os serviços prestados visam resolver um problema de um cliente.

Dois das equipes de desenvolvimento observadas trabalham em ambientes do tipo dedicado, uma é de ambiente autônomo e uma trabalha sob encomenda.

Sob o aval das três empresas, a cada uma das quatro equipes de desenvolvimento foi associado um observador. O objetivo dos observadores era identificar como o processo de desenvolvimento de software desenvolve-se em cada projeto. Depois disso, as características comuns dos vários processos deveriam ser identificadas para que, de alguma forma, fosse estabelecida uma relação entre a ocorrência de interações e as etapas do desenvolvimento de software. Para isso, os observadores deveriam, além de entender o processo, identificar e descrever quais as interações humanas que estão presentes nas fases do ciclo de vida do projeto. A Figura 1 apresenta as etapas que constituem o ciclo de vida de um projeto de software.

2.1. Observações de Acordo com o Ciclo de Desenvolvimento de Software

Tendo como base as observações feitas nas equipes e as etapas básicas do desenvolvimento de software (Figura 1), a primeira fase do processo de desenvolvimento de software é o levantamento ou o estudo das necessidades da aplicação, também conhecida como etapa de *análise dos requisitos*. Esta atividade, normalmente, começa quando o usuário solicita a informatização, parcial ou total, de suas atividades. Aqui, são identificados, além dos usuários responsáveis, um "escopo" inicial do sistema e as atuais deficiências no ambiente do usuário. Neste primeiro passo, também são estabelecidos os principais objetivos que o novo sistema deve possuir. Em [Fig95], são apresentados de forma mais detalhada os principais objetivos da atividade de levantamento de requisitos. Seu principal objetivo é a obtenção das informações necessárias e suficientes para o esboço de uma especificação inicial do sistema a ser desenvolvido, ocorrendo o levantamento das necessidades da aplicação, o levantamento detalhado do ambiente e o levantamento dos eventos externos aos quais o sistema deve reagir, pesquisa de mercado [Maff92].

A segunda fase observada corresponde à *especificação de projeto*. Aqui é feita a definição e a composição da arquitetura do software, mostrando o que construir e como construir. Inclui a decomposição do sistema em módulos, definição de estrutura de dados, modelagem dos dados e importantes definições algorítmicas (análise funcional).

Na fase de *implementação*, cada módulo do sistema deve ser especificado em detalhes, codificado, compilado (se for o caso), testado e, eventualmente, otimizado. Esta fase consiste, de acordo com [Maff92], na construção de um modelo expresso numa linguagem de representação que possa ser automaticamente convertida em uma linguagem de representação

¹ Este artigo foi escrito tendo como base, principalmente, o trabalho individual apresentado em [Fig95], onde encontram-se maiores informações sobre o estudo do trabalho cooperativo no processo de desenvolvimento de software.

inteligível para o hardware. É nesta terceira etapa do ciclo de desenvolvimento do software que ocorre o maior número de interações humanas, pois há uma comunicação quase que constante entre os membros da equipe. Estas interações são tipicamente de curta duração e muito dinâmicas. Tais características dificultaram suas observação e posterior modelagem por parte dos observadores.

Os testes correspondem à quarta fase que, inicialmente, engloba os testes de cada módulo construídos durante a fase de implementação e a posterior integração dos mesmos em uma estrutura única. Aqui ocorrem os testes como um todo, assegurando que os módulos se combinem e funcionem conforme foi projetado. Os testes beta, onde são enviadas cópias preliminares à usuários finais, também participam desta fase.

A quinta fase diz respeito à instalação e é também conhecida pelo nome de fase de implantação. Aqui ocorre a liberação da versão final do sistema que entra, então, em produção.

O processo de manutenção corresponde à última fase do ciclo de vida de um sistema de software. Nesta fase ocorre a revisão, modificação, conversão, aperfeiçoamento e/ou depuração do sistema já implantado em uma empresa. Aos poucos, o volume de modificações necessárias ao software cresce e dá lugar a uma nova fase de análise de requisitos, seja para o desenvolvimento de uma nova versão do sistema, seja para o desenvolvimento de um novo sistema.

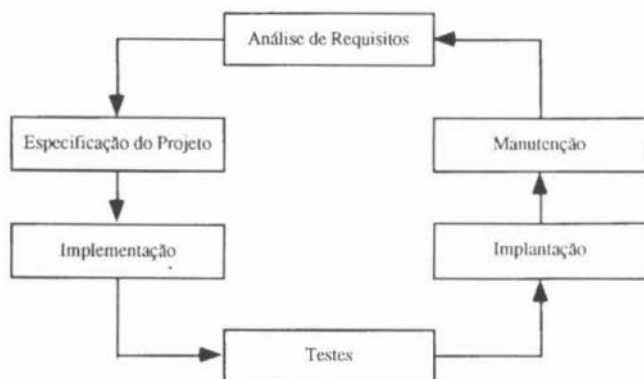


Figura 1 - Ciclo de Desenvolvimento de Sistema de Software

Tendo como base estas seis etapas e a informação de que o desenvolvimento de sistemas de software é feito segundo uma sequência não linear, tentou-se, durante as observações feitas nos grupos, classificar as interações humanas dentro das etapas do ciclo de desenvolvimento apresentado na Figura 1. Assim, verificamos que determinadas interações humanas acontecem sempre dentro da(s) mesma(s) fase(s). Além disso, verificou-se, também, que os resultados de certas interações determinam desvios e retornos no ciclo de desenvolvimento, influenciando, assim, em todo o processo.

As fases de análise de requisitos, especificação de projeto e implementação de sistemas de software foram os objetivos principais de nosso estudo, pois o nosso tempo estava bastante limitado e, coincidentemente, o estado dos projetos observados encontravam-se dentro das três fases acima.

2.2. A classificação das Interações Humanas segundo Martin Dürr

As interações humanas foram inicialmente classificadas em [Woit91] e [Dürr93] com base no trabalho cooperativo de grupos de pessoas. Abaixo, segue uma lista das principais situações onde há troca de informação no contexto do trabalho conjunto entre pessoas:

Afirmção;	Anúncio;	Aprovação;
Brainstorming;	Comunicação ou Aviso;	Confirmação;
Conversaão;	Delegação;	Discussão;
Escolha;	Imposição;	Marcação (ex.: de reunião)
Negociação;	Ordem ou Comando;	Pedido ou Pergunta;
Proposta;	Recusa;	Resposta;
Rompimento;	Sorteio;	Votação.

A partir destas situações, Dürr abstraíu as classes de interações que respresentam ações dentro dos planos de trabalho. Estas classes são caracterizadas por serem pontos de execução de funções de interação e de funções de coordenação. Assim, estas ações devem ser entendidas pelo leitor como pontos no plano de trabalho onde deve acontecer um contato entre membros do grupo, como por exemplo, para uma votação, discussão ou delegação de tarefas.

Inicialmente foram identificadas duas subclasses do processo interativo conforme sua função:

- Processo interativo informativo, onde representam as interções que realizam a simples troca de informações ou a tomada de conhecimento por várias pessoas do grupo. Aqui encontramos as figuras do emissor e do receptor;
- Processo interativo coordenativo. É o tipo de processo que representa as interações de tomada de decisão. Este tipo de processo pode ainda ser calssificado em:
 - Interação coordenativa preparativa para tomada de decisão, onde serem geradas possíveis alterantiva de decisão. Sua principal característica é a coleta e/ou preparação de material;
 - Interação coordenativa de tomada de decisão, ou seja, a partir da oferta de alternativas é feita a escolha por uma ou mais delas.

Quanto à tomada de decisão, Dürr identifica duas estratégias que poderão ser representadas no plano de trabalho: a primeira estratégia é a semântica E (*and*), que identifica que no momento da tomada de decisão, todas alternativas devem ser conhecidas; já a segunda estratégia, a semântica OU, diz que pelo menos uma das alternativas deve ser conhecida.

Uma apresentação mais detalhada sobre as classes de interações de Martin Dürr pode ser encontrado em [Dürr93].

2.3. A classificação das Interações Humanas segundo o Nosso Modelo

Ao aplicarmos a taxonomia apresentada por Dürr na modelagem dos fluxos de trabalho sendo observados, notamos que seus conceitos tornaram o modelo um tanto complexo e fora da realidade observada. Além disso, encontramos algumas ambigüidades em alguns símbolos apresentados por Dürr. Por exemplo, o símbolo de uma interação do tipo "preparação para tomada de decisão" (ex.: brainstorming) tem três arcos saindo de um losângulo. Estes arcos podem ser entendidos de duas maneiras diferentes, ou seja, eles podem representar um conjunto de alternativas geradas durante a interação ou estes arcos podem ser somente os elementos que podem ser usados para relacionar aqueles símbolos de interações aos passos seguintes no workflow. Este aspecto é melhor apresentado em [Ioch95].

Modificou-se, então, tanto as classes propostas em [Dürr93] como a notação gráfica a elas associada, com o fim de adequar o modelo de interações humanas à realidade observada no estudo de campo.

A notação gráfica conta com os seguintes símbolos:

- símbolos de interações;
- símbolos de atividades, onde são representadas as atividades que são executadas por uma única pessoa, sem interação;
- linhas de direção;
- símbolo de agregação de atividades;
- símbolo de agregação de interações.

A Figura 2 ilustra as três classes de interações humanas consideradas em nosso estudo. A primeira classe é a chamada interação informativa que representa formas de disseminação de informação. Este tipo de interação pode representar todos os processos interativos de informação, tais como, solicitação, questionamento, esquema de marcação de reuniões, confirmação e apresentação de projetos ou idéias. A classe das interações de preparação para tomada de decisão representa atividades tais como brainstorming, discussão e negociação. Neste tipo de interação, são geradas alternativas para uma posterior tomada de decisão. A classe das interações de tomada de decisão ilustra a opção por uma ou mais alternativas apresentadas num conjunto de possibilidades. Aqui ocorrem interações do tipo votação, imposição e decisão por consenso.



Figura 2 - As três classes de interações humanas representadas no modelo adotado

Cada uma das interações ilustradas na Figura 2 pode ser, ainda, classificada em *autônoma* (contém a letra "A" no interior do retângulo) ou em *coordenada* (mostram a letra "C" no interior do retângulo). Caso a interação seja autônoma, então ela não possui nenhum coordenador formal, podendo, contudo, haver lideranças informais. As interações coordenadas, por outro lado, apresentam coordenação formal, ou seja, possuem a presença da figura de um coordenador.

Outra sub-classificação que pode ser aplicada às três classes interações humanas diz respeito ao meio de comunicação (media) utilizado. Assim, as interações podem ser classificadas como sendo do tipo *face-a-face* (letra "F") ou *remotas* (letra R). O primeiro tipo representa interações onde os participantes dividem o mesmo espaço ao mesmo tempo (ex.: sala de reuniões). Interações remotas são as que permitem que os participantes estejam localizados, geograficamente, distantes uns dos outros (ex.: salas diferentes). Correio eletrônico pode ser utilizado para estabelecer interações remotas assíncronas (parceiros participam em tempos diferentes). Já um telefone pode ser o meio usado em interação remota síncrona (onde os interlocutores participam ao mesmo tempo).

Além de interações humanas, o modelo de descrição de fluxo de trabalho suporta atividades simples e complexas (Figura 3). Uma atividade simples é aquela desempenhada por um único membro da equipe de desenvolvimento. Uma atividade complexa representa, no modelo, a abstração de um subgrafo de atividades e interações do fluxo de trabalho. Tais atividades, ou somente interessam no conjunto, ou serão detalhadas à parte.

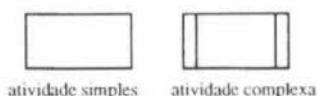


Figura 3 - Símbolos para Atividades

A linguagem gráfica, um dos elementos do modelo utilizado, apresenta, também, um símbolo para representar a direção e o sentido do fluxo de trabalho (Figura 4). As linhas de direção expressam dependência temporal entre atividades e interações humanas. Graças a esses símbolos, aspectos como paralelismo e ciclos podem ser representados no diagrama de workflow.

Usa-se o símbolo de agregação de interações para representar sua concomitância (Figura 4). Já a agregação de atividades e/ou interações é utilizada para que o conjunto resultante de ações seja tratado de forma monolítica. Este artifício é interessante quando, por exemplo, deseja-se representar um ciclo envolvendo um conjunto atômico de atividades e interações (Figura 5).

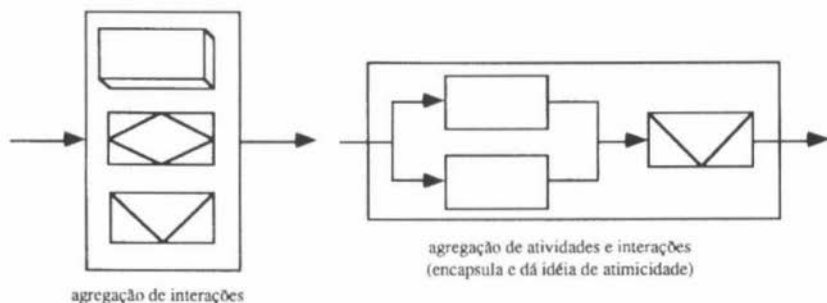


Figura 4 - Exemplo de agregação de interações e agregação de atividades

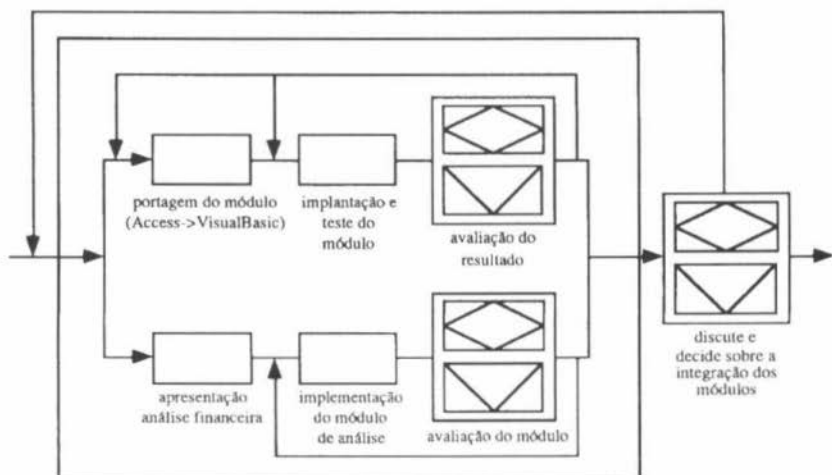


Figura 5 - Exemplo de fluxo de trabalho representado através do modelo de interações e atividades

Para complementar a descrição gráfica de interações humanas, o modelo proposto oferece uma linguagem textual. Através desta última, pode-se indicar aspectos operacionais e psicológicos de cada interação modelada no fluxo de trabalho. Ou seja, as interações modeladas, de início graficamente, são detalhadas, num segundo momento, através de uma

descrição textual que encerra as seguintes informações:

- Descrição da motivação (objetivos) da interação;
- Descrição dos papéis que estão envolvidos na interação;
- Como foi desenvolvida a interação (tipo de marcação de reunião, quem marcou, se foi confirmada ou não, quais as principais etapas, se houve coordenação formal ou não, se houve algum tipo de registro);
- Características psicológicas da interação (ex.: conflito de linguagem, clima de tensão ou desconfiância entre participantes);
- Resultado da interação.

3. Resumo dos Resultados Obtidos

Após a modelagem dos processos observados, passou-se para a segunda fase da pesquisa, ou seja, tabulação de informações estatísticas, tendo como base as informações coletadas junto às quatro equipes de desenvolvimento de software. As tabulações completas destes resultados e a metodologia utilizada para a realização desta tabulação podem ser encontradas em [Figu95].

O trabalho foi subdividido em dois pontos principais: identificação dos aspectos comuns a todas as observações e associação de ocorrências de interações a etapas do processo de desenvolvimento de software. O primeiro ponto diz respeito a uma tentativa de generalizar nossas observações para qualquer processo de desenvolvimento de software realizado por qualquer equipe. O segundo ponto refere-se à identificação de *interações típicas* de cada fase do processo de desenvolvimento. Era também nosso objetivo identificar possíveis interações que se repetem em diversas fases do processo.

De acordo com as modelagens feitas pelos membros da equipe de observação, temos as seguintes observações: o tipo de marcação das reuniões varia de telefone à marcação verbal (somente uma empresa apresenta utilização de mail eletrônico); a grande maioria das interações foram feitas face-a-face; quanto aos registros foram, na sua maioria, informais; mais do dobro das reuniões deveriam ser confirmadas; a grande maioria das interações apresentaram coordenação do tipo formal. A tabela 1 ilustra a tabulação total dos aspectos psicológicos relacionados com as interações.

3.1. Resultados Globais

De uma maneira geral, independentemente do tipo de interação ou da fase em que ela ocorreu, observou-se os seguintes aspectos operacionais e psicológicos.

3.1.1. Aspectos Operacionais

A maioria das interações combinadas com antecedência tiveram marcação via telefone. Além disso, a maioria das interações de preparação e tomada de decisão foram realizadas face-a-face. Isso demonstra que ainda é precário o uso de redes de computadores entre as equipes de desenvolvimento para tratar do andamento de projetos.

Outra observação que nos pareceu importante foi a de que, na maioria das interações, foram feitos registros de maneira informal. Não existe, por parte das equipes, a consciência da importância de documentar alternativas, argumentos, idéias, críticas, etc sobre o projeto. Mais ainda, a documentação feita não era integrada à documentação do projeto, ficando, portanto, fora do alcance de outros membros da equipe. Por fim, a documentação não era realizada utilizando o computador. Todavia, todas as equipes colocaram a necessidade de preservar as informações e decisões levantadas nas interações em um sistema integrado e de fácil acesso, que acompanhasse todo o processo de desenvolvimento e gerasse, por fim, sua documentação completa.

3.1.2. Aspectos Psicológicos

Durante as observações realizadas, a equipe tentou, também, captar o *clima psicológico* no qual as interações aconteciam. Por exemplo, interações em que predominava um clima de insegurança ou hostilidade, normalmente, produziam resultados (ex.: alternativas, decisões) de baixa qualidade devido ao baixo número de alternativas apresentadas. Isto tinha influência direta na produtividade da equipe e na qualidade (ex.: correção, performance) do produto final, visto que poderia haver negligência da melhor solução que resolveria determinado problema.

A tabela 1 apresenta um resumo dos climas predominantes nas interações observadas. De um total de 65 interações, somente 27 (41,5%) ocorreram em clima considerado normal. 13,8% transcorreram em clima de insegurança e 9,2% apresentaram diferença de linguagem. A insegurança pode ser explicada pelo fato de a maioria das interações terem apresentado coordenação formal. Na presença do chefe imediato ou do diretor da empresa, funcionários podem se sentir inseguros para apresentar, livremente, idéias e críticas, prejudicando, assim, a qualidade do software em desenvolvimento.

Já o problema de diferença de linguagem ocorreu mais ligado a interações entre o pessoal da equipe de desenvolvimento e o(s) usuário(s) do sistema. Culturas e jargões diferentes levam, frequentemente, a erros de compreensão.

Tipo de Clima	Ocorrências
normal	27
diferença de linguagem	06
insegurança	09
imposição	01
hostilidade	01
tensão	05
medo	01
insatisfação	05
normal e insegurança	02
normal e insatisfeito	02
conflito e insegurança	01
não informado	05
<i>total de ocorrências</i>	<i>65</i>

Tabela 1: Clima psicológico nas interações observadas

3.2. Dados Computados para Cada Tipo de Interação

Nesta seção, relaciona-se as interações modeladas às respectivas fases do ciclo de desenvolvimento de software onde foram observadas.

3.2.1 Interações Informativas

Nas interações informativas, o tipo de marcação mais usado foi o telefone, totalizando, aproximadamente, 50%. Outros meios usados foram o verbal, o verbal/telefone/e-mail, o verbal/telefone. Quanto ao tipo de confirmação, mais de 74% das marcações exigiam confirmação.

As maioria das interações informativas foram realizadas face-a-face. Mais de 50% delas não apresentaram coordenação formal e somente 22% delas eram registradas formalmente contra

66% registradas pelos participantes, por conta própria (ou seja, informalmente).

Normalmente, quem participava das interações informativas eram usuários, analistas de negócios e analistas de sistemas. Outro fato observado é que, geralmente, os usuários encontravam-se fora das empresas e, por esse motivo, as reuniões eram feitas por telefone ou oralmente. Não era utilizada rede de computadores pois as empresas (do usuário e da equipe) não estavam interligadas.

A insuficiência de software, no mercado, para registrar a história dos projetos pode ser a explicação da pouca ocorrência de registros formais feitos pelos grupos de desenvolvimento. Isso faz com que muitas informações sejam perdidas em pedaços de papel, sem um mínimo de organização.

No que diz respeito aos aspectos psicológicos das interações informativas, constatou-se que somente 30% delas foram realizadas sob um clima considerado normal. 13% destas interações apresentaram conflito de linguagem (ou seja, usuários e analistas não conheciam o mesmo vocabulário, dando diferentes significados para as mesmas palavras). Em [Ioch95], o autor explica que esse tipo de conflito deve-se ao fato de que pessoas de diferentes ramos de negócio podem ter diferentes culturas. Outro aspecto importante, destacado em [Ioch95], é que, atualmente, existem muito poucas ferramentas de suporte a situações de conflito de linguagem.

Outros aspectos psicológicos, tabulados sobre este tipo de interação, mostram-nos que um percentual de 8,15% delas foram realizadas sob um clima tenso, 4,3% apresentaram hostilidade. A insegurança estava presente em 17,4% das ocorrências e, como se já não bastasse, 17,4% das interações informativas apresentaram insatisfação por parte de seus participantes. Todos estes dados levam-nos a inferir que esses grupos são sérios candidatos a problemas com a produtividade e a qualidade do software por eles desenvolvidos.

3.2.2. Interações para Preparação de Tomada de Decisão:

A maioria absoluta das interações deste tipo foram realizadas face-a-face. Mais de 70% delas teve coordenação formal e todas as marcações foram devidamente confirmadas, sem no entanto, recorrerem ao computador para isso.

Um fato que nos surpreendeu muito foi o baixo número de reuniões documentadas de maneira formal (um total de 13%), visto ser esta classe de interação decisiva no processo de desenvolvimento de sistemas de software. Através de interações de preparação para tomada de decisão, usuários, analistas de negócios e analistas de sistema discutem, juntos, os requisitos que o software deve suprir e propõem alternativas de solução para suprir tais requisitos.

Quanto aos aspectos psicológicos, um terço das interações desta classe apresentaram insegurança (ex.: os projetistas não expunham com facilidade suas idéias e/ou críticas, pois tinham medo de represálias por parte da direção das empresas). Apenas um quinto das interações de preparação desenvolveu-se normalmente. Neste tipo de interação ainda estava presente o problema da diferença de linguagem.

3.2.3. Interações de Tomada de Decisão:

Todas as interações de tomada de decisão foram mantidas face-a-face. Algumas delas eram utilizaram o correio eletrônico. 28% das interações não foram formalmente documentadas e 39% delas não tiveram qualquer tipo de coordenação. Mesmo assim, as interações desta classe são as mais organizadas. As equipes de desenvolvimento têm consciência da importância das decisões tomadas para o sucesso ou fracasso do projeto.

O que mais nos chamou atenção neste tipo de interação foi o fato de que, pelo menos, 61% das interações humanas de tomada de decisão foram desenvolvidas sobre um clima normal. Já

para as interações de preparação para tomada de decisão foram tabulados apenas 30% neste clima. Em [Ioch95], também encontramos uma tentativa de explicação para este fato, ou seja, as pessoas poderiam já ter chegado a um consenso ainda antes do início de tais interações.

3.3. As Interações Relacionadas às Fases do Processo de Desenvolvimento de Software

Na fase da análise dos requisitos, as interações eram longas e muito mais formais se comparadas com as outras. Aqui ocorre, principalmente, diferença de linguagem, a documentação é feita de maneira mais formal e as interações são face-a-face. Além disso metade das interações ocorreram informalmente. Quanto a confirmação, 70% foram confirmadas. Pode-se observar, ainda que, esta fase é caracterizada por interações com clara coordenação.

Alguns problemas foram observados nesta fase, tais como: o usuário não dispunha de muito tempo para as reuniões, a marcação das reuniões era difícil, perdia-se muito tempo procurando anotações feitas informalmente nas reuniões; faltava objetividade no desenvolvimento das reuniões.

A fase de especificação do projeto também apresentou o problema da diferença de linguagem. Os registros continuavam a ser feitos informalmente. As interações continuavam longas.

A fase caracterizada por micro-interações, ou seja, pequenas interações que ocorrem de forma constante, muitas vezes em conjunto com outras, é a fase de implementação. Nesta fase os participantes interagiam quase que constantemente uns com os outros, seja em forma de perguntas, discussões, sugestões, decisões, trocas de objetos [Livi92], comunicações ou esclarecimentos.

4. Principais Requisitos para Auxiliar os Desenvolvedores de Sistemas

Como o leitor pode ter percebido, foram verificadas várias dificuldades no processo de desenvolvimento de software. Abaixo serão descritos alguns dos tantos problemas encontrados por nós durante esses quatro meses de observações.

O primeiro problema observado foi o tempo em que os sistemas são desenvolvidos (produtividade), ou seja, os usuários mostravam-se, muitas vezes, insatisfeitos por ocorrer atrasos nos projetos. Um dos motivos destes atrasos é a falta de documentação, pois a maioria das empresas mantém as suas informações documentadas precariamente, muitas vezes, de maneira informal.

Um segundo motivo para a baixa produtividade dos projetos é a falta de coordenação formal das interações, ou seja, as reuniões são geralmente feitas com pouco ou quase nenhum tipo de coordenação, ocasionando tumultos, falta de objetividade e atrasos em decisões.

O processo de desenvolvimento de sistemas de software implica na necessidade de tomada de decisão. As decisões deveriam ser tomadas com rapidez, segurança e correção, procurando-se, sempre, levar em conta o maior número de alternativas de solução possíveis.

Outro problema encontrado é a sub-utilização dos computadores. A maioria das empresas não utiliza o computador para marcar seus compromissos, discutir assuntos, ou, até mesmo, enviar simples mensagens (interações informativas).

Entretanto, o principal problema encontrado nas empresas foi a diferença de linguagem. O mesmo vocabulário possui significado diferente para as pessoas do grupo de desenvolvimento. Isso dificulta consideravelmente o processo de comunicação entre os membros da equipe.

O compartilhamento de objetos também constitui importante fator no atraso dos projetos, pois um objeto criado por determinado projetista pode ser necessário a outro. Este problema poderia ser resolvido se as equipes utilizassem algum sistema gerenciador de objetos e trocas [Livi92].

Seria interessante ainda que os projetistas contassem com a ajuda de algum tipo de simulador, capaz de prever, antecipadamente, o comportamento dos programas, antes destes serem implementados. Talvez o teste de uma especificação formal do software evitasse idas e voltas custosas nas fases de projeto e implementação.

Por último, um gerente de workflow seria bem vindo, pois, assim, o desenvolvimento de sistemas de software teria um fluxo pré-determinado, diminuindo, com isso, a tendência que os projetistas têm em realizar passos desnecessários ou esquecer de passos importantes para o bom desenvolvimento de sistemas.

Assim, diante da lista de problemas apresentada acima e da tabulação estatística das observações feitas junto às equipes de desenvolvimento, inferiu-se o que acreditamos ser os principais requisitos para auxiliar no processo de desenvolvimento de software:

- gerência/sistema de documentação que acompanhe todas as fases do processo;
- suporte à coordenação de reuniões e outras interações;
- gerência/sistema de comunicação entre projetistas e usuários;
- suporte à tomada de decisão;
- auxílio à solução de problemas de diferença de linguagem;
- suporte ao compartilhamento de objetos;
- gerência do workflow;
- suporte à simulação no processo de desenvolvimento de sistemas de software;
- gerência de editores compartilhados.

5. Análise das Principais Ferramentas Disponíveis no Mercado que Poderiam Auxiliar os Projetistas no Desenvolvimento de Seu Trabalho

A Tabela 2 resume os resultados da análise das mais populares ferramentas de desenvolvimento de software que suportam algum tipo de trabalho cooperativo². Para cada software analisado, foi investigado se ele supre os requisitos apresentados no Capítulo 4. O símbolo "+" indica que a ferramenta supre o requisito expresso na coluna. O símbolo "-" indica que, ou a ferramenta não supre o requisito, ou não foi possível encontrar informação confiável a este respeito. Para efeito de economia de espaço, cada requisito foi associado a um número de um algarismo. Ao analisar a Tabela 2, o leitor deve, portanto, fazer as seguintes associações:

Gerência de documentação	->	1;	Gerência de comunicação:	->	5;
Coordenação de interações	->	2;	Compartilhamento de objetos:	->	6;
Suporte à tomada de decisão:	->	3;	Edição compartilhada:	->	7;
Auxílio à diferença de linguagem:	->	4;	Gerência de workflow:	->	8.

² Como a maioria das equipes observadas não utilizavam qualquer tipo de ferramenta que as auxiliassem durante o desenvolvimento de seus trabalhos, não foi possível informar o número de pessoas que utilizam determinada ferramenta.

A descrição completa da análise das ferramentas que constam da Tabela 2 mais a de outras ferramentas pode ser encontrada em [Fig95].

Nome da Ferramenta	1	2	3	4	5	6	7	8
Novell GroupWise 4.1 [Exam94]	-	-	-	-	+	-	-	-
Lotus Notes [Exam94]	-	-	-	-	+	-	-	-
Link Works [Exam94]	-	-	-	-	+	+	-	-
Exchange [Exam94]	-	-	-	-	-	-	-	-
Documents [Exam94]	-	-	-	-	+	+	-	-
Notes Express [Exam94]	-	-	-	-	+	+	-	-
ForComment [Exam94]	-	-	-	-	-	-	+	-
FormFlow [Rodd92]	-	-	-	-	-	-	-	+
KeyFile [Rodd92]	-	-	-	-	-	-	-	+
FlowMark [Rodd92]	-	-	-	-	-	-	-	+
Person to Person [Rodd92]	-	+	-	-	+	-	-	-
CoNex [Hahn90]	+	-	+	-	-	+	+	-
GRACE [Benf92]	+	-	-	-	-	-	-	-
Higgins [Oppe88]	-	-	-	-	+	-	-	+
The Coordinator [Oppe88]	-	-	-	-	+	-	-	-
Caucus [Oppe88]	-	+	+	-	-	-	-	-
Office Work [Oppe88]	-	-	-	-	+	-	-	-
Life [Oppe88]	-	-	-	-	+	-	-	-
Informations Lens [Crow88]	+	-	-	-	+	-	-	-
Object Lens [Crow88]	+	-	-	-	-	-	-	-
Quorum [Arau94]	+	-	+	-	-	-	-	-
SDSE [Elli88] ³	+	-	-	-	-	+	-	+
Callisto [Elli88]	-	-	-	-	-	-	-	-
COKES [Elli88]	-	-	-	-	-	-	-	+
Polymer [Elli88]	-	-	-	-	-	-	-	-
Argnoler [Elli88]	-	+	+	-	+	-	-	-
ACT! [Elli88]	-	-	-	-	-	-	-	-

Tabela 2: Suporte aos requisitos dado pelas ferramentas existentes

Observando-se a Tabela 2, pode-se notar, entre outros aspectos, que nenhuma das ferramentas analisadas supre todos os requisitos identificados através do trabalho de campo. Apenas um dos 27 softwares listados suporta 4 dos oito requisitos levantados (CoNex). A maioria supre um ou dois requisitos e existem ferramentas que não atendem a requisito algum (ex.: Callisto).

O requisito mais atendido pelas ferramentas em geral é o de suporte à comunicação entre usuários (número 5). 44,4% dos sistemas analisados atendem a este requisito. Já os aspectos do processo de desenvolvimento de software mais negligenciados pelas ferramentas parecem ser o conflito de linguagem e a edição compartilhada. Nenhuma ferramenta suporta resolução à diferença de linguagem e somente duas delas (7,4%) se preocupam em auxiliar a edição de textos compartilhada.

³ System Development Support Environment.

6. Conclusões e Trabalhos Futuros

Este artigo apresentou a instrumentalização e os resultados de um estudo de campo, envolvendo a observação de quatro equipes de desenvolvimento de software. Através desta observação, inferiu-se uma série de requisitos que um ambiente de suporte ao desenvolvimento de software deveria suprir para garantir maior produtividade e qualidade tanto ao processo como ao produto final.

Em uma segunda parte do artigo, foram apresentados alguns dos resultados alcançados a partir da análise de um número significativo de ferramentas comerciais para suporte ao desenvolvimento de software. Tais ferramentas foram investigadas quanto o suporte que prestam ao processo em relação aos requisitos levantados a partir do estudo de campo. Notou-se que poucas ferramentas oferecem suporte a aspectos fundamentais do desenvolvimento cooperativo de software, como é o caso da diferença de linguagem e a editoração compartilhada.

Através desse trabalho espera-se que sejam desenvolvidas novas ferramentas que atendam, se não a todos, pelo menos a um conjunto significativo dos requisitos aqui levantados. Não estamos ainda bem certos se o melhor não seria a integração, no ambiente, de várias ferramentas, cada uma especializada no suporte a um subconjunto dos aspectos aqui tratados.

Como trabalhos futuros, tendo como base este estudo de quase um ano, pretende-se, criar um modelo que integre os mecanismos de gestão de interações (apresentados neste trabalho) com os mecanismos de gestão de trabalho cooperativo já propostos para o ambiente STAR [Wagn94].

7. Referências Bibliográficas

- [Arau94] Araujo, R.: Quorum: Um SSDG para Desenvolvimento de Software. In.: Simpósio Brasileiro de Engenharia de Software. Curitiba, outubro de 1994.
- [Benf92] Benford, S. et al.: An Information Sharing Approach to CSCW. In: Computer Communications, University of Nottingham, vol. 15, n°. 8, p. 502-508, outubro de 1992.
- [Crow88] Crowston, K. et al.: Cognitive Science and Organizational Design: A Case Study of Computer Conferencing. In.: Computer-Supported Cooperative Work: a book of Reading by Irene Greif. p. 713-740. San Mateo, 1988.
- [Dürr93] Dürr, M.: Coordination Mechanisms for Team-Work: Proposal of Both a Model and Database Support. PhD. Dissertation, University of Karlsruhe, Institut für Programmstruktur und Datenorganisation, Karlsruhe, Germany, 1993 (em Alemão).
- [Elli88] Ellis, C. et al.: Office Information Systems and Computer Science. In.: Computer-Supported Cooperative Work: a book of Reading by Irene Greif. p. 199-247. San Mateo, 1988.
- [Exam94] Maestros do Trabalho em Grupo. Exame Informática. Novembro de 1994.
- [Figu95] Figueiredo, E.: Estudo do Trabalho Cooperativo no Processo de Desenvolvimento de Software. Trabalho Individual n°. 475, Porto Alegre: CPGCC - UFRGS, Janeiro, 1995.
- [Fish90] Fisher, A. S.: CASE - Usando Ferramentas para o Desenvolvimento de Software Editora Campus, Rio de Janeiro, 1990.
- [Ioch95] Iochpe, C.: Understanding, Modeling, and Supporting The Software Development Process In South Brazil. Proc. of the Int. Workshop on Concurrent/Simultaneous Engineering Applications, Lisboa, 5-7 de abril de 1995.
- [Livi92] Livi, C.: Um Mecanismo de Gestão de Trabalho Cooperativo para o Ambiente de

- [Maff92] Maffeo, B.: Engenharia de Software e Especificação de Sistemas. Editora Campus, Rio de Janeiro, 1992.
- [Oppe88] Opper, S.: A Groupware Toolbox. Byte, p. 275-282. December, 1988.
- [Rood92] Rodden, T. et al.: Distributed System Support for Computer Supported Cooperative Work. Computer Communication. Lancaster University, vol. 15, n°. 8, p. 527-538, outubro de 1992.
- [Thio87] Thiollent, M.: Metodologia da Pesquisa-Ação. Cortez Editora, São Paulo, 1987.
- [Wagn94] Wagner, F. R.; Golendziner, L. G.; Fornari, M. R.: A Tightly Coupled Approach to Design and Data Management. Proc. of the European Design Automation Conference, Grenoble, France, 1994.
- [Woit91] Woitass, M.: Coordination in Structured Conversation: A Coordination Model for Cooperating Agents and Its Use in CSCW. Oldenburg-Verlag, Munich, 1991 (em Alemão).