

Modelagem da Interação do Usuário no Desenvolvimento Ágil

Cecília E. Giuffra¹, Patrícia Vilain¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

{giuffra,vilain}@inf.ufsc.br

Abstract. *Agile methods have been recently applied to software development as an alternative to traditional processes. Modeling the interaction between users and the system can be included in agile methods as an additional practice to promote the communication with users, thus resulting in benefits not only when communication is carried out but also during the rest of the development process. This paper analyses how such a practice could be included into the agile development.*

Keywords: *Agile Methods, Interaction Modeling, User Interaction Diagram.*

Resumo. *Dentre os processos de desenvolvimento de software, os métodos ágeis vêm sendo uma opção bastante explorada. Para auxiliar no atendimento do princípio de comunicação constante com o usuário, a modelagem da interação do usuário com o sistema pode ser incluída como uma das práticas dos métodos ágeis, trazendo benefícios tanto durante a comunicação como no resto do processo de desenvolvimento. Esse artigo faz uma análise de como a modelagem da interação pode ser realizada no desenvolvimento ágil.*

Palavras-chave: *Métodos Ágeis, Modelagem da Interação, Diagrama de Interação de Usuário.*

1. Introdução

Os métodos ágeis são processos de *software* bastante utilizados atualmente. Eles surgiram no final dos anos 90, como uma alternativa aos métodos tradicionais, e propõem agilizar o processo de desenvolvimento de *software*. Esses métodos possuem em comum o fato de serem aplicados em projetos de baixa complexidade, utilizando ciclos iterativos curtos, planejamento guiado por funcionalidades, retro alimentação constante, tolerância a mudanças, proximidade da equipe, intimidade com o cliente e um foco no ambiente geral de trabalho da equipe [Machado 2005]. Entre os métodos ágeis podemos citar: XP (*Extreme Programming*), Scrum, Crystal, FDD (*Feature Driven Development*), DSDM (*Dynamic Systems Development Method*).

Esses métodos são caracterizados por repetidas iterações durante o desenvolvimento do software. Cada iteração, que dura entre duas e oito semanas, apresenta a entrega de uma versão funcional para o cliente, onde é informado o que foi feito até o momento, com o intuito de intercambiar opiniões e idéias, tanto entre os programadores e projetistas como entre projetistas e clientes.

Os métodos ágeis oferecem às organizações uma grande variedade de práticas e enfoques para o desenvolvimento de *software*, propiciando que as organizações escolham o método que melhor supra suas necessidades. Apesar destes métodos serem guiados por funcionalidades, onde um conjunto de funcionalidades é desenvolvido a

cada iteração, eles as especificam de diferentes maneiras. Além disso, por não fazer parte dos princípios ágeis, a modelagem da interação entre o usuário e o sistema nem sempre é realizada nos diferentes métodos. Entretanto, para aplicações interativas, que podem ser vistas como aquelas que apresentam interação entre o usuário e o sistema [Preece 94], a modelagem desta interação poderia facilitar a comunicação entre o usuário e o desenvolvedor. E embora os métodos ágeis sejam muito utilizados em aplicações do tipo sistemas de informação que apresentam bastante interação com o usuário, eles não dão ênfase nesta modelagem.

Este artigo apresenta como é feita a modelagem da interação entre o usuário e o sistema nestes métodos ágeis, e analisa se existem vantagens na inclusão desta modelagem nestes métodos.

O artigo está organizado da seguinte maneira. A seção 2 apresenta o processo dos métodos ágeis e a forma como eles fazem a modelagem do usuário com o sistema. A seção 3 apresenta a extensão dos métodos ágeis estudados (FDD, DSDM, XP e Scrum) com a modelagem da interação. E, por fim, na seção 4 são apresentadas as conclusões.

2. Métodos Ágeis e a Interação do Usuário

No geral, os processos dos métodos ágeis estão divididos basicamente em uma parte de planejamento e uma parte iterativa, onde se realiza o desenvolvimento da aplicação em si. Na parte de planejamento são feitos, geralmente, os estudos do sistema, a documentação necessária, o cálculo dos custos do projeto, a criação das histórias de usuário ou lista de funcionalidades do sistema, entre outros. Na parte iterativa é desenvolvido o projeto, baseando-se na lista de funcionalidades ou histórias de usuário feitas na etapa de planejamento, e em alguns casos, como o XP, são realizados testes de aceitação e unidade antes da implementação.

Todos os métodos tem um conjunto de práticas parecidas, em alguns casos em maior número que em outros, mas no geral todos os métodos ágeis incluem a comunicação constante com o usuário e a entrega de versões funcionais no final de cada iteração. Porém, a modelagem da interação do usuário com o sistema não é uma prática adotada por esses métodos. De acordo com [Vilain 2002], a análise e modelagem dessa interação é importante para identificar a informação manipulada por um sistema computadorizado e a funcionalidade que este sistema deve oferecer. Normalmente a interação é descrita de forma textual através de casos de uso, mas esta interação também pode ser representada por notações diagramáticas, tais como, diagramas de transição de estados, diagramas de seqüência, diagramas de colaboração, diagramas de atividades, *sketches*, etc, apesar destas técnicas não apresentarem como objetivo a representação da interação [Vilain 2002].

3. Extensão dos Métodos Ágeis

Como os métodos ágeis não apresentam explicitamente uma prática para a modelagem da interação do usuário com o sistema, alguns métodos foram analisados e foi proposta a inclusão desta modelagem nestes métodos com o intuito de verificar vantagem da utilização desta modelagem. A modelagem da interação do usuário com o sistema foi inserida no início da parte iterativa de cada um dos processos. A escolha desta posição dentro do processo foi feita porque em cada iteração podem ser adicionados novos

requisitos, os quais podem requerer uma nova modelagem da interação para ser mostrada ao cliente antes da implementação. Assim, cada requisito será analisado para verificar se ele tem troca de informação ou não e, assim, se precisa ou não desta modelagem.

A seguir é mostrado como esta nova prática pode ser utilizada no desenvolvimento ágil através da sua inclusão em quatro métodos ágeis distintos.

3.1. Processo FDD

O FDD (*Feature Driven Development*) é um método ágil para gerenciamento e desenvolvimento de software, criado em 1997 por Peter Coad e Jeff De Luca, num grande projeto em Java para o United Overseas Bank, em Singapura. Ele “combina as melhores práticas do gerenciamento ágil de projetos com uma abordagem completa para Engenharia de Software orientada por objetos, conquistando os três principais públicos de um projeto de software: clientes, gerentes e desenvolvedores.” [Heptagon]

O processo do FDD está dividido em cinco etapas que são: desenvolver um modelo abrangente, construir a lista de funcionalidades, planejar por funcionalidade, projetar cada funcionalidade e construir por funcionalidade. Dessas etapas as três primeiras pertencem à parte de planejamento do processo e as duas últimas à parte iterativa. Nesse método a prática de modelagem será inserida dentro da parte iterativa, antes do projeto de cada funcionalidade, conforme mostrado na Figura 1. Para cada funcionalidade associada com uma grande troca de informações entre o usuário e o sistema, a interação é modelada. Assim, poderá ter-se uma visão mais clara e específica do comportamento de cada requisito. A modelagem nesse método pode ser realizada pelo programador chefe. Essa prática é adicionada às existentes.

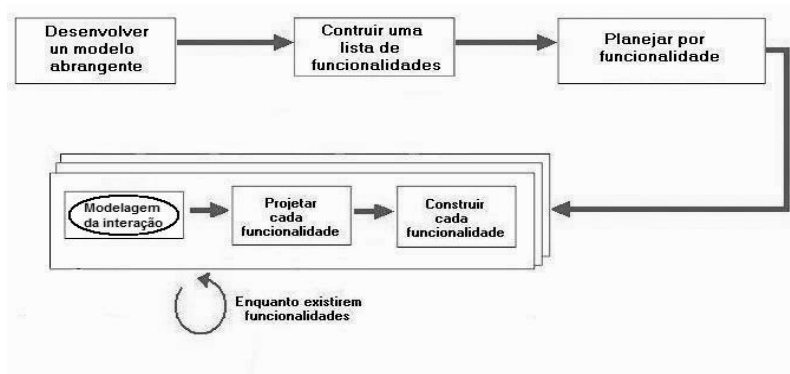


Figura 1. Modelagem da Interação no FDD

3.2. Processo DSDM

O DSDM (*Dynamic System Development Method*), desenvolvido na década dos 90 na Inglaterra, é um framework para o desenvolvimento ágil de software, portanto uma de suas características é a aceitação de mudanças. Ele é uma extensão do RAD (*Rapid Application Development*) e tem como objetivo principal gerenciar as ações dentro dos limites desejados a respeito do tempo e do orçamento [Teixeira].

O processo do DSDM apresenta três etapas: Pré-projeto, Projeto e Pós-projeto. O projeto, por sua vez, está dividido em cinco fases: estudo de viabilidade, estudo de negócio, modelagem funcional, projeto e construção, e implementação. A parte iterativa

nesse método se encontra dentro do projeto, durante a execução das três últimas fases dessa etapa. Nesse método a prática de modelagem da interação substituirá a criação dos protótipos e poderá ser feita dentro da fase do projeto, na etapa de modelagem funcional, quando são desenvolvidos os protótipos que serão mostrados ao cliente, conforme mostrado na Figura 2. Sempre que os requisitos selecionados na iteração corrente tratarem de interações entre o usuário e o sistema, a modelagem da interação do usuário com o sistema deverá substituir os protótipos. A modelagem da interação no DSDM pode ser realizada pelo gerente do domínio. A prática de prototipagem é substituída pela modelagem da interação.

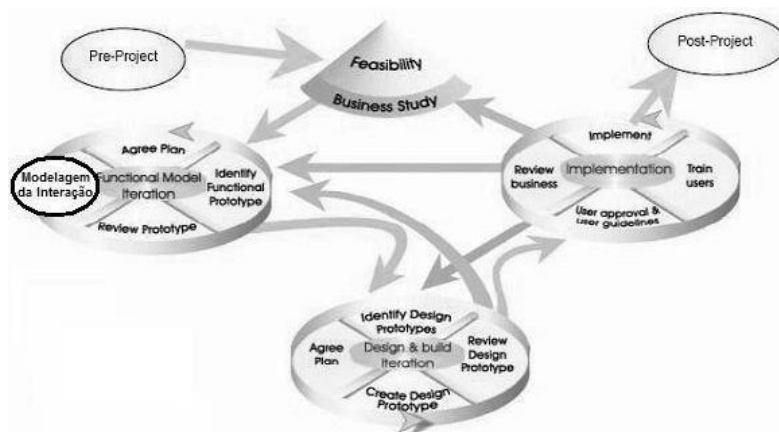


Figura 2. Modelagem da Interação no DSDM

3.3. Processo XP

O XP (*Extreme Programming*), criado por Kent Beck no final da década de 90, é o método ágil mais conhecido atualmente. Assim como outros métodos ágeis, entre suas características podemos citar: grupos pequenos de desenvolvedores; softwares desenvolvidos em um tempo menor do que os desenvolvidos utilizando métodos tradicionais; requisitos dinâmicos que não estão totalmente definidos e que mudam constantemente [Fagundes 2005]. “No XP o cliente define as funcionalidades do sistema que será desenvolvido por meio das chamadas estórias do usuário, e as prioriza em seguida” [Garcia et al].

O processo do XP tem cinco etapas, sendo que somente a terceira corresponde à parte iterativa do método, embora as etapas de planejamento e produção possam ser executadas mais do que uma vez. Nesse método a prática de modelagem será adicionada antes de iniciar cada iteração, após a fase de planejamento, onde são escolhidas as estórias que vão ser usadas na primeira iteração, conforme mostrado na Figura 3. Para cada estória de usuário que apresenta grande troca de informações entre o usuário e o sistema, também será modelada a interação do usuário com o sistema. Isso ajudará na hora da comunicação com o cliente para que o cliente compreenda melhor a interação entre o usuário e o sistema, especificamente as informações trocadas na iteração. A modelagem da interação do usuário no XP pode ser realizada pelo testador. É proposta a inclusão da prática de modelagem da interação às práticas já existentes no método, e ela pode ser aplicada de forma conjunta com a prática jogo de planejamento, que é onde se decide o que vai ser feito na seguinte iteração.

3.4. Processo SCRUM

Scrum, cuja definição foi formalizada por Ken Schwaber nos anos 90, é um método ágil direcionado para a gerência de projetos. O objetivo desse método é a entrega, dentro de iterações, de um software de qualidade que seja útil para o cliente. Essas iterações são formadas por *Sprints*, que são ciclos de, no máximo, 30 dias, onde se trabalha para alcançar objetivos específicos. Estes objetivos são representados por uma lista de funcionalidades, chamada de *Product Backlog*, definida pelo Dono do Produto (*Product Owner*) e constantemente atualizada e priorizada [Sanchez].

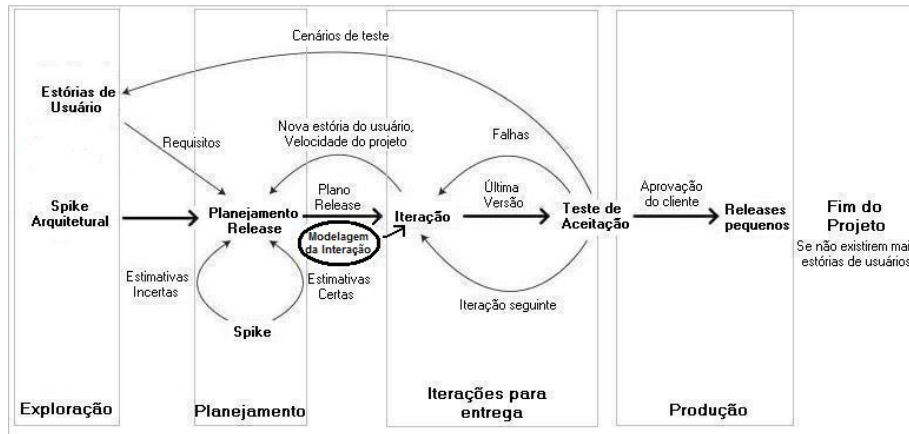


Figura 3. Modelagem da Interação no XP

O processo do *Scrum* está dividido em três etapas: *PreGame*, *Game* ou Desenvolvimento e *PostGame*. A parte iterativa é a de desenvolvimento. Nesse método a modelagem da interação do usuário será realizada na fase de desenvolvimento. após a criação do *Product Backlog* e antes da criação do *Sprint Backlog*, que é a lista das funcionalidades a serem desenvolvidas no próximo *Sprint*, ou próxima iteração, conforme mostrado na Figura 4. Para cada funcionalidade associada com uma grande troca de informações entre o usuário e o sistema, será modelada a interação do usuário com o sistema. Assim, o cliente poderá ver como vai funcionar a interação entre o usuário e o sistema segundo os requisitos que ele pediu. A modelagem da interação no Scrum pode ser realizada por quem define o *Sprint Backlog*. Essa modelagem, assim como no XP, também pode ser vista como uma nova prática no *Scrum*, e pode ser aplicada em conjunto com a prática *Sprint Backlog*.

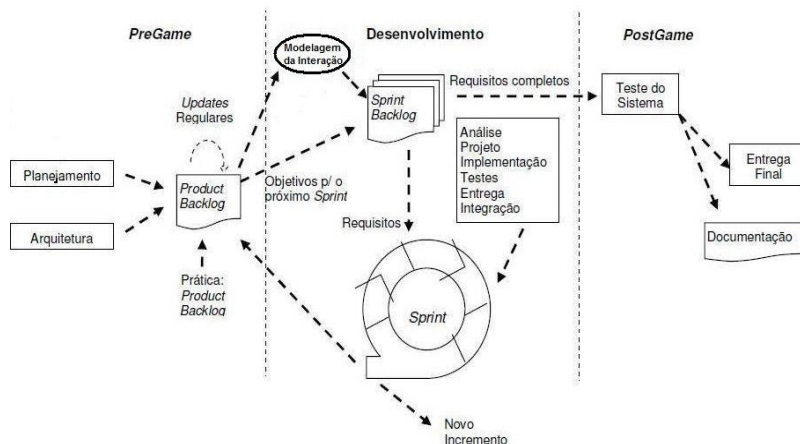


Figura 4. Modelagem da Interação no Scrum

4. Estudo de Caso

Para analisar a inclusão da prática de modelagem da interação no desenvolvimento ágil, os quatro métodos ágeis apresentados na seção anterior foram utilizados no desenvolvimento de duas aplicações. Cada aplicação foi desenvolvida usando dois métodos ágeis. Cada método ágil foi utilizado em duas iterações, sendo que em uma iteração foi utilizado o processo original e em outra iteração foi incluída a prática de modelagem da interação do usuário com o sistema.

As duas aplicações foram realizadas por um único desenvolvedor, tendo somente um cliente em cada aplicação, que era quem tinha o contato contínuo com o desenvolvedor, e, ocasionalmente, tinha outro usuário pertencente à empresa (cliente) que testava a versão funcional do *software* depois de cada iteração. Assim, no desenvolvimento dessas aplicações não foram aplicadas as práticas que precisavam mais do que uma pessoa na equipe, entre elas podemos citar as reuniões diárias do *Scrum*, a programação em pares do XP, as equipes por funcionalidade do FDD e o *workshop* do DSDM.

Para modelar a interação nos quatro métodos ágeis foi utilizada a técnica UID (*User Interaction Diagram*), que representa a interação do usuário com o sistema.

A seguir são apresentados os UIDs e o resultado do desenvolvimento de cada aplicação.

4.1. UIDs

Os UIDs foram propostos para representar a interação entre o usuário e o sistema, necessária para realizar as tarefas desejadas pelo usuário de uma aplicação [Vilain 2002]. Os UIDs podem ser usados em conjunto com os casos de uso e, portanto, podem ser usados em qualquer processo de software que os utilize. Para cada caso de uso pode ser definido um UID composto por um conjunto de estados de interação, que incluem as informações trocadas durante a interação usuário-sistema, conectados através de transições.

Na Figura 5, apresentada a seguir, é mostrado um UID com seus principais elementos: entrada do usuário (informação fornecida pelo usuário), estado (contém a informação trocada entre o usuário e o sistema), transição (troca de foco da interação), e saída do sistema (informação do sistema que é mostrada para o usuário).

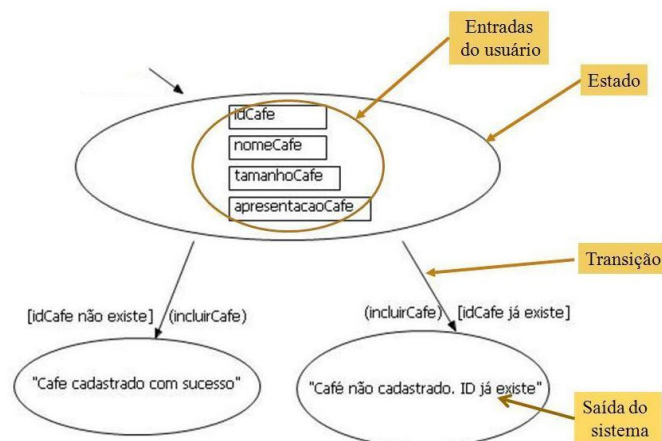


Figura 5. Exemplo de UID

Esse UID corresponde à funcionalidade de cadastro de um novo tipo de café, funcionalidade pertencente à primeira aplicação. Nesta interação o usuário fornece o identificador, nome, apresentação e tamanho do novo tipo de café que será adicionado. Para que essa funcionalidade seja concluída corretamente o identificador inserido não deve existir, pois todos os cafés têm identificador único.

4.2. Primeira aplicação: Sistema de Vendas Fábrica de Café

A primeira aplicação desenvolvida apresentou as seguintes funcionalidades: venda dos cafés, nas diferentes apresentações e tamanhos; relatório de estoque e vendas; registro do pedido, emissão de ordem de processamento; registro do pagamento; cadastramento e exclusão de cadastro de cliente; inclusão de novo café e novo estoque; exclusão de café. Nesta aplicação foram usados os métodos FDD, nas duas primeiras iterações, e DSDM, nas duas seguintes. A Figura 6 mostra a interface inicial desta aplicação.

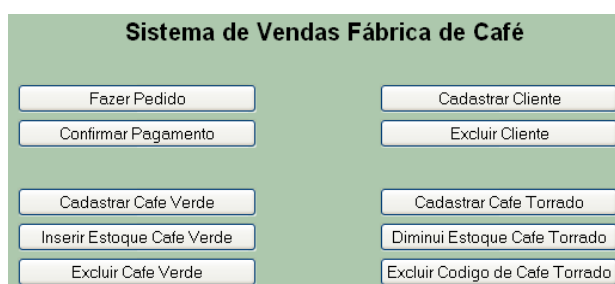


Figura 6. Interface inicial do sistema

4.2.1. Análise do FDD com Modelagem da Interação

Na primeira iteração o processo do FDD foi desenvolvido normalmente, sem modificações, já na segunda iteração foram incluídos os UIDs no início da iteração.

Nesse método, os diagramas de seqüência e o diagrama de classes foram suficientes para realizar a análise de projeto. Nesse caso, pelas características do método, que entre os métodos ágeis analisados neste trabalho, é o mais parecido com os métodos tradicionais, e requer a criação de vários diagramas, os UIDs foram utilizados durante a comunicação com o usuário e não no desenvolvimento do projeto.

Pode-se notar a utilidade da modelagem da interação entre o usuário e o sistema, pois o cliente/usuário conseguiu entender facilmente os diagramas mostrados, o que manteve a comunicação ágil. Em relação ao tempo não foi feita uma medição exata, mas a inclusão/criação dos UIDs não significou uma diferença notória para o desenvolvedor, cujo papel foi desenvolvido por uma pessoa só, e foi quem assumiu também o papel de responsável pela criação dos UIDs. Não se notou essa diferença, pois os UIDs são diagramas fáceis de desenhar. Além disso, os UIDs também são fáceis de entender pelo usuário, que terá um conhecimento maior do produto desenvolvido, no momento da comunicação entre ele e o desenvolvedor, o que é uma vantagem, pois a comunicação fica mais proveitosa.

4.2.2. Análise do DSDM com Modelagem da Interação

Na terceira iteração, com o DSDM, os protótipos foram realizados usando os UIDs, e na quarta iteração foi executado o processo normal do método utilizando as telas de interfaces ao invés dos UIDs.

No DSDM os UIDs foram aproveitados de uma forma diferente do que no FDD, pois foram usados substituindo os protótipos do sistema, que podem ser feitos com código ou com as telas da interface. Na terceira iteração os UIDs substituíram os protótipos de telas da interface e mostraram a interação usuário-sistema, mantendo a agilidade e sendo mais úteis do que o uso somente das telas pois eles servem também para modelar a interação do usuário. Já no caso de protótipo de código, não foi desenvolvida nenhuma iteração com este tipo de protótipo, mas provavelmente a agilidade no seu uso seria menor do que no uso dos UIDs, pois os UIDs não requerem a programação. É importante frisar que os UIDs só são indicados para substituir protótipos que mostram a interação entre o usuário e o sistema. Em relação ao tempo, assim como no método anterior, não foi realizada nenhuma medição exata, mas percebeu-se que o uso dos UIDs substituindo os protótipos de telas é um pouco mais rápido.

4.3. Segunda aplicação: Sistema Administrativo de gastos da empresa.

As funcionalidades da segunda aplicação incluíram: cadastro de folha de pagamento de funcionário; cadastro, atualização e relatório de despesas; relatório de vendas, relatório de receitas; cálculo do saldo mensal. Nesta aplicação foram usados os métodos XP, nas duas primeiras iterações, e Scrum, nas duas seguintes. A Figura 7 mostra a interface de uma das funcionalidades do sistema.

Atualiza Funcionario:

CPF: (só números)

CPF:

Nome:

Endereço:

Telefone:

Email:

Tipo:

Salario:

Horas Extra:

Faltas:

Figura 7. Interface atualiza funcionário

4.3.1. Análise do XP com Modelagem da Interação

Na primeira iteração o processo do XP foi desenvolvido sem a inclusão de UIDs. Os UIDs foram utilizados no início da segunda iteração desenvolvida com o XP.

No XP, que não requer a criação dos diagramas da UML, e onde são criados testes de unidade e aceitação antes do desenvolvimento do sistema, os UIDs serviram de apoio como entrada para a implementação tanto do código quanto da interface. Além disso, o XP não possui uma modelagem da interação entre o usuário e o sistema, e por ser um método onde a comunicação com o usuário é contínua, a inclusão dos UIDs no XP melhorou o entendimento do usuário. Em relação ao tempo, apesar de não ter sido feita uma medição exata, a iteração onde os UIDs foram utilizados não apresentou uma demora notória para o desenvolvimento da aplicação.

4.3.2. Análise do *Scrum* com Modelagem da Interação

Na terceira iteração, desenvolvida com o *Scrum*, foi utilizado o processo original desse método. Somente na quarta iteração é que os UIDs foram inseridos no início da parte iterativa, ou seja, na etapa de desenvolvimento ou *GamePhase*.

O *Scrum* é um pouco diferente dos outros métodos porque ele não define um procedimento para o desenvolvimento, sendo mais voltado à gerência do projeto. A inserção dos UIDs no *Scrum*, que também não requer a criação de qualquer diagrama UML, serviu de apoio para a implementação do código, pois os UIDs simulam a parte da interface. Entretanto, a inclusão dos UIDs dependerá do processo usado dentro das *Sprints*. Se for usado um processo que já modela a interação entre o usuário e o sistema, os UIDs não serão necessários.

5. Conclusões

Neste trabalho foi feita uma análise da necessidade da modelagem da interação entre o usuário e o sistema no desenvolvimento ágil, tomando como exemplo os métodos ágeis FDD, o DSDM, o XP e o *Scrum*. Para isto foi realizado um estudo de caso com duas aplicações que foram desenvolvidas utilizando estes quatro métodos incluindo a modelagem da interação. Neste estudo de caso os UIDs foram usados como técnica de modelagem da interação entre o usuário e o sistema, entretanto qualquer técnica direcionada para esta modelagem poderia ter sido utilizada.

Os métodos ágeis que foram usados são bastante diferentes entre si. O FDD é um método um pouco mais parecido com os métodos tradicionais; o DSDM utiliza a prototipação no desenvolvimento do *software*, prática que foi substituída pela técnica de modelagem da interação; o XP que é mais direcionado para a programação; e o *Scrum* é direcionado para o gerenciamento de projetos e não especifica um processo a ser seguindo dentro de cada iteração.

Durante a aplicação destes quatro métodos, sentiu-se a falta de uma modelagem que permitisse mostrar ao usuário o comportamento do sistema no momento da sua interação com ele. E como os métodos ágeis têm entre suas características a comunicação constante e a opção de adicionar, a qualquer momento do desenvolvimento da aplicação, algum requisito que o usuário possa ter esquecido, pode-se notar a importância desta modelagem para que essa comunicação seja melhorada.

No estudo de caso desenvolvido, a modelagem da interação foi realizada através da definição de um UID para cada funcionalidade ou estória do usuário, e pode-se notar que no caso de funcionalidades com mais troca de informações, os UIDs foram bastante úteis, já que em sistemas com funcionalidades com pouca troca de informação a modelagem não faz diferença. Em relação ao tempo, o tempo dispendido na criação dos UIDs não significou muito em comparação ao que eles oferecem na comunicação com o usuário.

Outra constatação importante foi que a modelagem da interação foi mais aproveitada em alguns métodos do que em outros. No FDD não fez muita diferença a inclusão desta modelagem no processo. No caso do DSDM esta modelagem foi usada no lugar dos protótipos, deixando o processo com igual ou maior agilidade. No XP, que não requer a criação dos diagramas clássicos e onde são criados testes de unidade e aceitação antes do desenvolvimento do sistema, a modelagem da interação serviu de

apoio como entrada para a implementação tanto do código quanto da interface. No *Scrum*, que também não requer a criação de qualquer diagrama clássico, a inclusão da modelagem da interação facilitou um pouco a implementação do código, pois os UIDs simularam a parte da interface com o usuário.

Independente do método utilizado, o uso dos UIDs para modelar a interação do usuário com o sistema é uma maneira simples e direta de se comunicar com o usuário. Isto se faz de uma maneira rápida, pois o usuário consegue entender o fluxo de dados sem muito problema. Portanto, a inclusão da modelagem da interação pode ser vista como uma prática opcional que pode ser incluída em qualquer método ágil sempre que uma funcionalidade que apresenta grande interação entre o usuário e o sistema estiver sendo desenvolvida. Já em aplicações que tem pouca interação esta prática não é muito útil.

Apesar do estudo de caso ter sido desenvolvido com uma equipe de apenas um desenvolvedor, isto não invalida a análise feita neste trabalho porque as práticas ágeis relacionadas com a modelagem da interação foram executadas. Além disso, o estudo de caso realizado para esse trabalho mostra o melhor lugar onde a prática da modelagem pode ser inserida, sem prejudicar a agilidade dos métodos e facilitando a comunicação com o usuário ou cliente, que na maioria das vezes não tem conhecimento prévio do desenvolvimento de um sistema.

6. Bibliografia

- Fagundes, P. B. (2005) “Framework Para Comparação e Análise de Métodos Ágeis”. Dissertação de Mestrado, Depto Informática e Estatística, UFSC.
- Garcia D.C, E. et al. “Padrões e Métodos Ágeis: agilidade no processo de desenvolvimento de software”. Universidade Federal de São Carlos – Departamento de Computação. Universidade de São Paulo – Instituto de Ciências Matemáticas e de Computação.
- Heptagon. “FDD - Feature Driven Development”. Disponível em: <http://www.heptagon.com.br/fdd> – Último acesso em 18/03/2009.
- Machado, T. L. (2005) “Uma Ferramenta de Suporte ao *Framework* para comparação e Análise de métodos Ágeis”. Trabalho de Conclusão do Curso Sistemas de Informação, Depto Informática e Estatística, UFSC.
- Preece, J. et al. (1994) *Human-Computer Interaction*. Addison-Wesley, 775p.
- Sanchez, I. “*Coding Dojo Floripa* Desenvolvimento Ágil”. Disponível em: <http://dojofloripa.wordpress.com/2007/02/07/scrum-em-2-minutos/> - Último acesso em: 24/05/09.
- Teixeira, D. et al. “DSDM – Dynamic Systems Development Methodology”. Faculdade de Engenharia da Universidade do Porto. Disponível em: http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_14.pdf - Último acesso em: 27/05/09.
- Vilain, P. (2002) “Modelagem da interação com o usuário em aplicações hipermídia”. Tese de Doutorado, PUC-RIO, Rio de Janeiro.