METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS

Keila de Carvalho Freitas[1], Demétrio Renó Magalhães[2], Francisco Antonio Fernandes Reinaldo[3]

Abstract — *This article presents a comparison between two software development methodologies, such as Typical Software Development Methodology and Fast Software Development Methodology. In order to compare both methodologies, all development and documentation steps of the Time-Tabling Scholar Collaborative System (SCIHE) were carefully measured. Results showed that Fast Software Development Methodology proved to be faster, flexible and offer an up-to-date documentation process for short projects.*

Index Terms — Fast Methodology, Typical Methodology, MRDS, Object Oriented Development

INTRODUÇÃO

Método é um procedimento ou técnica utilizado para obter um determinado resultado. Também pode ser definido como uma forma de agir em uma determinada situação a fim de regularizar uma tarefa. Com essa definição é possível entender o porquê é fácil encontrar várias metodologias para o desenvolvimento de documentação de software. Isso acontece porque são vários os tipos de software que são desenvolvidos em várias linguagens para diversos ambientes e desenvolvidos em diversos modelos de sistemas de informação, tais como baseado em fluxo de dados, relacionais e orientado a objetos.

Os projetos podem ser classificados em grande, médio e pequeno porte. Isso vai depender de diversas variáveis relacionada na Tabela 1.

TABELA 1
CLASSIFICAÇÃO DE PROJETOS [PASSOS, MARIA LUÍZA, 2008]

Característica	Classificação		
	Baixo	Médio	Alto
Duração	< 6 meses	6-18 meses	> 18 meses
Pressão por prazos	Redução de até 5% do prazo	Redução de 5 – 20% do prazo	Redução de mais de 20% do prazo
Custos	Até 1 milhão na moeda de referência	Entre 1 milhão e 10 milhões na moeda de referência	> 10 milhões na moeda de referência
Incerteza Tecnológica	Tecnologia amplamente dominada pela organização	Tecnologia não dominada pela organização e de poucos fornecedores no mercado (até três fornecedores)	Tecnologia desconhecida (a desenvolver) pela organização e pelo mercado
Instabilidade de Escopo	Mudanças de até 5% no escopo	Mudanças entre 5 e 20% no escopo	Mudanças de mais de 20% no escopo
Escopo do Gerenciamento	Até 3 departamentos e/ou 3 empresas envolvidas	Entre 3 e 9 departamentos e/ou empresas envolvidas	Acima de 9 departamentos e/ou empresas envolvidas
Riscos	Alterações até 5% no custo, prazo ou qualidade.	Alterações entre 5 e 20% no custo, prazo ou qualidade	Alterações acima de 20% no custo, prazo ou qualidade.
Valor para o negócio (riscos de negócios)	Baixo	Médio	Alto
Nível de mudanças organizacionais	Baixo	Médio	Alto

A documentação é uma forma de definir quais as atividades serão desenvolvidas no decorrer do projeto; quando, por quem e como serão executadas essas tarefas; controlar e verificar o andamento do projeto; padronizar o desenvolvimento e tratar possíveis problemas que aparecem no decorrer do desenvolvimento. As atividades típicas de um processo de desenvolvimento são agrupadas em vários passos, a serem realizadas na Metodologia típica de Desenvolvimento. A metodologia a ser abordada e comparada com a metodologia típica é a Metodologia Rápida de Desenvolvimento de Sistemas (MRDS), muito utilizada para a construção de documentação para sistemas de pequeno e médio porte. Essa metodologia tem o propósito de diminuir a quantidade de documentos produzidos na construção do sistema, facilitando o desenvolvimento e tornando-o mais rápido e simples. Essa metodologia é

1 Keila de Carvalho Freitas, Aluna de Computação – Sistemas de Informação do Centro Universitário do Leste de Minas Gerais. Av. Tancredo Neves, 3500, B. Universitário, Coronel Fabriciano, Minas Gerais, Brasil, keilafreitas@gmail.com.
2 Prof. M.Sc. Demétrio Renó Magalhães, Professor Orientador do artigo. Membro do GPET (Grupo de Pesquisa em Educação e Tecnologia). Centro Universitário do Leste de Minas Gerais. Av. Tancredo Neves, 3500, B. Universitário, Coronel Fabriciano, Minas Gerais, Brasil, demetriorm@gmail.com.
3 Prof. M.Sc. Francisco Antonio Fernandes Reinaldo, Professor Coordenador do LIC – Laboratório de Inteligência Computacional. Membro do GPET (Grupo de Pesquisa em Educação e Tecnologia). Centro Universitário do Leste de Minas Gerais. Av. Tancredo Neves, 3500, B. Universitário, Coronel Fabriciano, Minas Gerais, Brasil, reinaldo.opus@gmail.com.

composta por duas etapas que são: Modelo de Requisitos onde se define todos os requisitos do sistema; e o Modelo de Análise e Projeto onde ajudará o projetista a organizar e estruturar suas decisões.

ATIVIDADES TÍPICAS DE UM PROCESSO DE DESENVOLVIMENTO

Existem várias formas de modelar um sistema, cada forma com suas fases, com momentos diferentes e diversas ênfases. Mas algumas atividades são comuns a maioria das metodologias, essas atividades são:

- Levantamento de Requisitos;
- Análise de Requisitos;
- Projeto;
- Implementação;
- Testes;
- Implantação.

O Levantamento de Requisitos refere-se ao entendimento do negócio da organização, entender quais são as regras de negócios da organização, quais são suas falhas, o que deve ser melhorado, o que o sistema irá abordar dessas regras e suas soluções. Nessa etapa são levantados os requisitos do sistema. Para que isso seja feito é necessário entrevistas dos analistas com o cliente e com os usuários finais, quem realmente trabalha com o sistema. Além das entrevistas é recomendada a leitura de manuais, observação das atividades dos usuários finais e pesquisas de sistemas similares já existentes no mercado.

O produto do Levantamento de Requisitos é a Documentação de Requisitos que descreve os requisitos funcionais, não-funcionais e restrições. Nessa etapa não são cobrados requisitos técnicos do sistema, tais como definir a linguagem do sistema, a plataforma entre outros. O objetivo dessa etapa é descrever o que o sistema deve ter para atender as necessidades do usuário.

A Análise de Requisitos chamada também de especificação de requisitos é o momento em que o analista estuda os requisitos levantados na etapa anterior para que a solução seja a mais adequada para o problema. Nessa etapa os requisitos técnicos continuam não sendo importantes. O foco é identificar a melhor solução. No sistema orientado a objetos, o produto dessa fase será a construção das classes e dos objetos.

A próxima etapa é o Projeto. Nessa etapa será decidida qual linguagem usar, qual banco de dados será adotado e especificar um padrão de programação entre os desenvolvedores.

O Projeto pode ser dividido em duas etapas, de acordo com Bezerra (2002,25:26):

1. Projeto de Arquitetura ou Projeto de Alto Nível que, considerando um projeto orientado a objetos, consiste em organizar as classes de objetos em subsistemas e seus componentes.
2. O Projeto Detalhado ou Projeto de Baixo Nível é o projeto de interface, para ser apresentado ao cliente, e o projeto do banco de dados.

A etapa de implementação consiste em codificar tudo aquilo que foi decidido e documentado na etapa de Projeto. No projeto orientado a objetos também fazem parte dessa etapa a construção de classes.

Após a implementação do sistema é necessário fazer os testes. Essa etapa não acontece separadamente das outras, isso porque a todo o momento o programador do sistema realiza testes no sistema. É importante nessa etapa a construção de um relatório de testes e erros como um relatório de históricos.

METODOLOGIA RÁPIDA DE DESENVOLVIMENTO DE SISTEMAS (MRDS)

Durante o desenvolvimento de um sistema utilizando essa metodologia é possível diminuir a quantidade da documentação. Para a construção dessa documentação o processo é dividido em duas fases que são:

- Modelo de Requisitos, nessa fase será identificado os requisitos que o sistema deverá cumprir. Essa fase é muito importante para a qualidade do projeto; e
- Modelo de Análise e Projeto, nessa segunda fase o projetista construirá uma solução a partir das análises e nos requisitos predefinidos na primeira fase.

No Modelo de Requisitos serão construídos os casos de uso do sistema, a interface ou protótipos e as classes. Essa constituição pode ser observada no Meta - Modelo abaixo (Schmitz, E. A. e Silveira, D. S. 2001, 7).

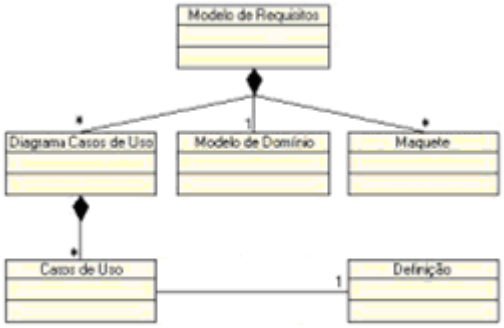

FIGURA 1
META-MODELO DO MODELO DE REQUISITOS [SCHMITZ, E. A. E SILVEIRA, D. S., 2001]

Para a produção do Modelo de Requisitos são necessárias algumas etapas, apresentada por Schmitz, E. A. e Silveira, D. S. (2001, 7:8):

Em forma de texto explicar qual o objetivo do sistema, como ele funciona atualmente, sem o sistema, quais as suas falhas, seus problemas. Essa é a definição do sistema; Construir as classes, seus métodos e atributos. Essas classes serão melhoradas no decorrer do desenvolvimento do

sistema e no momento de modelagem dos casos de uso, mas é importante construí-lo nesse momento; Nesse passo devem ocorrer várias atividades em conjunto. Devem-se fazer: os protótipos do projeto; definir e listar os atores do sistema e os casos de uso; estruturar os casos de uso, ou seja, definir extensões, interfaces e quais e como serão reutilizados de alguma forma; decidir a organização das classes referentes ao pacote; O último passo é consolidar as informações extraídas até o momento e verificar se as afirmações da interface desenvolvida nos protótipos estão em sintonia com os casos de usos definidos. Caso não esteja é nessa fase que realizarão as correções e melhorias para que essa sintonia ocorra.

É importante ressaltar que o levantamento de requisitos feita nessa fase precisa ser bem elaborado para a sua construção. Caso contrário as alterações se tornam tão freqüentes durante o desenvolvimento que o atraso pode ser ainda maior se caso estivesse utilizando uma metodologia típica de desenvolvimento de sistemas.

O Modelo de Analise e Projeto é o módulo em que será codificado para a construção do projeto, conforme o modelo a seguir segundo Schmitz, E. A. e Silveira, D. S. (2001, 8):

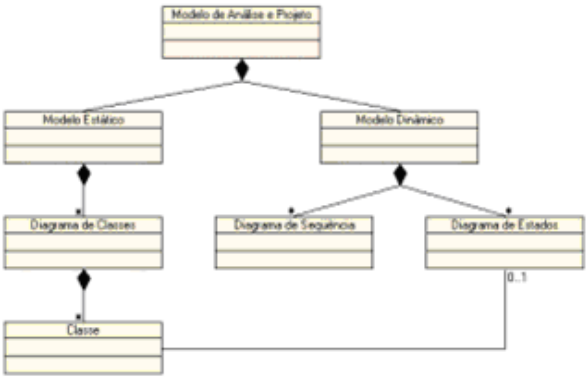

FIGURA 2
META-MODELO DO MODELO DE ANÁLISE E PROJETO [SCHMITZ, E. A. E SILVEIRA, D. S., 2001]

Esse modelo de desenvolvimento de software é interativo, é cíclico, ou seja, é necessário executar esses passos várias vezes até obter o resultado final esperado.

Para a execução desse modelo é preciso os seguintes passos conforme mostrado por Schmitz, E. A. e Silveira, D. S. (2001, 8:9):

O primeiro passo é definir os diagramas de classes e mostrar a interação entre classes, protótipos e controles de cada pacote definido; O próximo passo é construir os diagramas de seqüência para cada pacote. Esse passo é muito importante para que posteriormente possamos construir os casos de uso do sistema; Logo em seguida é importante que sejam construídos diagramas de transição de estados para classes, domínios, protótipos e controles cujo ciclo de vida seja incomum; Pesquisar e analisar as classes construídas e verificar se não pode ser simplificado, ou herdado ou ainda reutilizados os códigos para facilitar na codificação do sistema; Por ultimo revisar e certificar-se que não há nada para melhorar e corrigir.

No desenho abaixo, Schmitz, E. A. e Silveira, D. S. (2001, 9), é possível entender essas duas fases do processo MRDS e entender como ele pode ser cíclico.

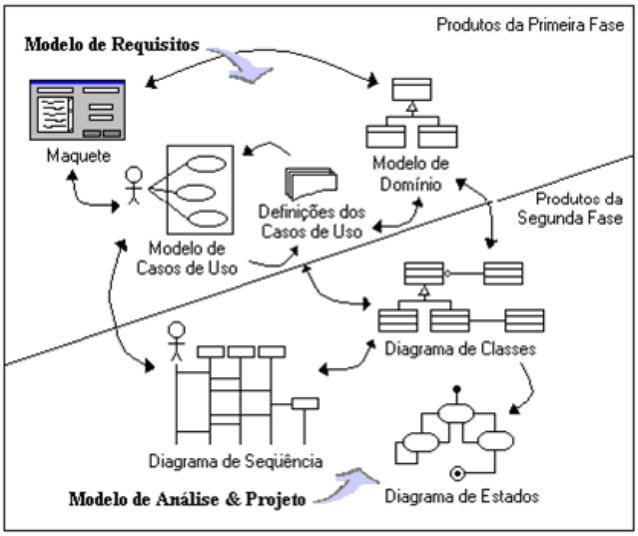

FIGURA 3
DIAGRAMA ESQUEMÁTICO DA MRDS [SCHMITZ, E. A. E SILVEIRA, D. S., 2001].

Cada uma das metodologias de desenvolvimento abrange atividades diferentes, de formas diferentes, para projetos diferentes e logo obtemos resultados diferentes.

A metodologia MRDS é desenvolvida para projetos orientados a objetos e de médio e pequeno porte. Isso facilita a interpretação e a construção do projeto devido a sua disposição de melhoria contínua sem interferir no tempo do projeto. A documentação, a análise e o projeto evoluem juntos e de acordo com a necessidade da organização, deixando os participantes da construção preocupados apenas em resolver da melhor maneira os problemas do sistema.

Podemos definir 3 aspectos importantes nessa metodologia:

1. Os objetos do sistema, que estão preocupados com os dados do sistema, são encontrados e definidos na fase de Modelagem de Requisitos;
2. A interação entre usuário final e sistema, são os protótipos e eventos externos e internos que promovem essa interação;
3. E o controle do sistema, onde se encontra as camadas do sistema, a sequência dos acontecimentos do sistema entre interface, domínio e controle.

COMPARAÇÕES

Para que seja escolhida uma metodologia é preciso conhecer algumas características do projeto. Analisando o Projeto de Iniciação Científica, o Sistema Colaborativo

Inteligente de Horário Escolar (SCIHE), é possível classificá-lo como pequeno porte.

O quadro a seguir mostra a comparação entre as duas metodologias de desenvolvimento estudas nesse artigo:

TABELA 2
COMPARAÇÃO DAS METODOLOGIAS

Atividades típicas de um processo de desenvolvimento	Metodologia Rápida de Desenvolvimento de Sistemas (MRDS)
Levantamento de Requisitos são dados sobre as necessidades, sobre o negócio da empresa. Nessa etapa tem-se a documentação com requisitos funcionais, não-funcionais e restrições.	Os requisitos são levantados na primeira fase com a definição dos requisitos funcionais, não-funcionais e restrições.
Análise de Requisitos é realizada pelo analista do sistema, momento em que serão validados os requisitos e tomadas decisões em cima dele.	São analisados na primeira fase pelo analista com o Modelo de Domínio, Modelo de Casos de Uso e os Protótipos.
No Projeto é decidido sobre linguagens a serem usadas e cria-se os protótipos para serem apresentadas ao cliente.	É projetado na segunda fase com o Diagrama de Seqüência e Diagramas de Estados, ainda nessa etapa é permitido a alteração de requisitos sem que o projeto se atrase.
Implementação é a construção das classes juntamente com a codificação do projeto.	É Implementado na segunda fase com a construção das classes, do controle e da interface e o próprio projeto.
Os Testes são realizados em toda a implementação do sistema, e são realizados vários tipos de testes.	É testado na segunda fase, onde são construídas as classes e o projeto, porém os testes são realizados em todos os momentos da segunda fase.
A Implantação é realizada após os testes com o cliente, onde ele valida o sistema e são realizadas novas manutenções.	É implantado na segunda fase, após o término de vários ciclos das fases para que o projeto fique consistente.

Essa comparação veio para esclarecer diferenças entre as duas metodologias e esclarecer como a MRDS consegue trabalhar todas as fases de uma metodologia típica, sem perder nenhuma etapa, mas sendo realizadas ao mesmo tempo em que o projeto será construído.

CONCLUSÃO

A partir desse artigo, foi possível conhecer algumas metodologias e esclarecer que a Metodologia Rápida de Desenvolvimento de Sistemas se enquadra perfeitamente para o SCIHE, pois ela permite a evolução e correção do sistema no decorrer do desenvolvimento, o que permite maior flexibilidade para profissionais com pouca experiência em análise, maior liberdade para alterar decisões feitas no início do projeto, sem que isso interfira diretamente no tempo de execução do projeto.

Ou seja, será possível fazer alterações em requisitos do projeto durante a construção dele, sem que isso tenha como conseqüência uma perda significativa no tempo e gastos excessivos com o projeto. Mas como toda análise, é preciso que seja feita com muita clareza e responsabilidade, pois uma vez feita uma análise defasada, esse tempo ganho no projeto utilizando o MRDS será desperdiçado com correções em excesso e haverá um atraso muito maior no projeto, tornando a construção do sistema inviável.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 1.ed. Rio de Janeiro: Elsevier, 2002.

[2] DEITEL, H. M.; DEITEL, P. J. Java: como programar; tradução e revisão técnica Carlos Arthur Lang Lisbôa. 4.ed. São Paulo: Bookman, 2003.

[3] FOWLER, M.; SCOTT, K. UML Essencial;. 2.ed. São Paulo: Bookman, 2000.

[4] PASSOS, Maria Luíza. Desburocratizando o Gerenciamento de Projetos. 2008. Disponível em: <http://www.pmies.org.br/v2/centraladm/artigos/arquivos/Jornal_2008_Jul_Word.pdf >. Acesso em: 24 de Março de 2009.

[5] SCHMITZ, E. A.; SILVEIRA, D. S. MRDS: Uma Metodologia de Desenvolvimento de Software em Empresas de Pequeno e Médio Porte. Portugal, 2001.

[6] SOMMERVILLE, Ian. Engenharia de Software. 6.ed. São Paulo: Addison Wesley, 2003.