

Análise individual e comparação entre metodologias de desenvolvimento

Resumo:

Este texto pretende realizar uma comparação independente de uma experimentação prática de 3 metodologias de desenvolvimento usadas no mercado atualmente: XP, RUP e OOHDM.

Serão definidos critérios de comparação após uma breve análise individual e uma posterior análise de suas vantagens e desvantagens. Estes critérios serão usados então para a realização da comparação entre as mesmas.



Roberto de Holanda Christoph

Índice:

1.Introdução.....	3
2. Descrição individual.....	4
2.1 XP.....	4
2.2 OOHDM.....	8
2.3 RUP.....	10
3. Vantagens e desvantagens individuais.....	13
3.1 XP.....	13
3.2 OOHDM.....	14
3.3 RUP.....	15
4. Estabelecimento dos critérios de comparação.....	15
5. Comparação entre as metodologias.....	19
Bibliografia.....	28

1 – Introdução

O objetivo deste texto é prover uma comparação independente de experimentação prática entre metodologias de programação usadas no mercado atualmente.

A ausência de experimentação prática se deve ao fato de que para a realização da mesma, seria necessário gastar um tempo considerável para cada uma das metodologias escolhidas para a comparação, tempo este que não pode ser obtido devido ao prazo estipulado para a versão final deste documento.

As metodologias escolhidas são usadas atualmente por diversas empresas e possuem um número de seguidores fiéis. Todas elas já foram testadas diversas vezes pelo mercado, desta forma existe material amplamente disponível para a análise de cada uma delas individualmente tanto na internet quando na forma de material impresso (livros, teses, etc). São elas as metodologias XP (Extreme Programming), RUP (Rational Unified Process) e OOHDM(Object-Oriented Hypermedia Design Method).

Abaixo segue uma breve definição de cada uma das metodologias escolhidas:

- XP é basicamente um conjunto de regras, valores e princípios que irão permitir o rápido desenvolvimento de softwares sem comprometer sua qualidade, satisfazendo todos os requisitos do cliente de maneira satisfatória e em curto prazo [GI]. É conhecida como uma metodologia “leve”(Lightweight methodology) ou seja, não envolve um conjunto muito grande de regras e princípios a serem seguidos rigidamente, sendo portanto uma metodologia fácil de se usar.
- RUP é uma metodologia de desenvolvimento orientada a objetos, e que provê guias, templates e exemplos para todos os aspectos e estágios do desenvolvimento de um programa [PK]. Foi feita pela Rational e possui integração com suas suítes. Uma característica bem marcante desta metodologia de desenvolvimento é o fato de possuir uma estrutura bem definida e regular.

- OOHDM é um modelo de base para o desenvolvimento de aplicações hipermídia que norteia as etapas de construção de um projeto e o uso da tecnologia relacionada. É composta por 4 atividades diferentes, projeto conceitual, projeto navegacional, projeto abstrato de interface e implementação [DSW].

Com o decorrer deste texto, será apresentado uma descrição mais detalhada de cada uma destas metodologias, descrevendo como elas atuam, o que pregam, suas regras e princípios. Após esta descrição será feito um levantamento individual de suas vantagens e desvantagens, focando unicamente na metodologia da vez. Em seguida teremos uma parte do texto dedicada ao estabelecimento dos critérios de comparação que serão usados, para por fim comparar cada uma das metodologias escolhidas com base nestes critérios pré-definidos, levando em conta domínios variados.

2 – Descrição individual

Este capítulo tem como objetivo uma descrição individual mais profunda de cada uma das metodologias escolhidas. Tal descrição será útil para que se tenha idéia de o que cada uma delas propõe e como propõe. Será feita uma descrição de sua história, seus princípios e valores.

2.1 – XP

Em 1996, Kent Beck começou um projeto em DaimlerChrysler usando novos conceitos em desenvolvimento de software, conceitos esses criados por ele no começo da década de 90 com ajuda de Ward Cunningham (pessoa com quem havia trabalhado). A esses novos conceitos deu-se o nome de metodologia de Extreme Programming (XP) [GI].

A metodologia tem grande ênfase em trabalho de equipe, são considerados membro de uma equipe tanto os programadores quanto gerentes

e principalmente o cliente, sendo que este terá sempre uma participação bem ativa junto a equipe de desenvolvimento.

A comunicação entre todos os membros da equipe é um fator essencial da metodologia XP, é através dela que será possível criar um design simples e eficiente para o projeto, possibilitando assim que este seja entregue ao cliente o mais cedo possível para que caso necessário sejam implementadas as mudanças nos requisitos solicitados.

Outro fator importante são os testes, já que estes são definidos antes da implementação em si e será através deles que será feita a avaliação de sucesso ou não de uma determinada parte do projeto.

Segue abaixo uma descrição da metodologia em si:

A primeira fase é conhecida como planejamento, nessa fase é necessário a cooperação de toda a equipe, existirá uma interação muito grande entre os gerentes, clientes e programadores.

Primeiramente é necessário a formulação das estórias(User Stories) e para tanto o cliente deve descrever cada função que ele deseje ver no produto final. A descrição deve ser feita em linguagem natural pelo cliente e não deve ser muito longa, sendo o ideal aproximadamente 3 linhas escritas em um cartão.

Após a elaboração das estórias é necessário que o cliente proponha testes para cada um destes. Esses devem ser simplesmente o que o cliente espera que o sistema responda devido a uma certa entrada. Esses testes serão importantíssimos, pois nenhuma estória pode ser considerada completa sem ter passado em todos os testes propostos.

Consequentemente as tarefas(reformulação das estórias em uma linguagem um pouco mais técnica) devem ser distribuídas pelos membros da equipe(não é necessário distribuir todas as tarefas, as que sobram serão divididas mais tarde) e a cada uma deve ser associado um tempo de desenvolvimento.

Devem ser feitas reuniões periódicas no decorrer do projeto, essas reuniões devem ocorrer sempre em um período de 1 a 3 semanas e serão o coração do projeto. Nestas reuniões serão discutidos os progressos de cada

tarefas e se necessário serão feitas reformulações nas divisões das tarefas bem como nos requisitos, a cada reunião serão feitas também as distribuições de novas tarefas para os programadores. Outro passo importante é a reunião onde serão definidas as datas de lançamento das versões do produto. Nesta reunião devem ser definidos quais estórias devem ser implementadas em cada lançamento e também as datas destes lançamentos.

Deve ser feito uma reunião informal todos os dias, onde serão discutidas as dificuldades encontradas. Essa reunião não deve ser muito longa e pode até mesmo ser feita em frente a um computador de um programador da equipe. O objetivo desta reunião é evitar que sejam feitas diversas outras durante o dia com um número reduzido de pessoas em cada uma delas, aumentando assim a produtividade.

É muito importante nesta fase de planejamento não ter medo de mudanças nos requisitos ou de reavaliação de prazos, já que isso é perfeitamente normal de ocorrer durante qualquer projeto [KB].

A próxima fase é conhecida como design, e a palavra chave para um bom design na metodologia XP é simplicidade.

Sempre tente fazer tudo do jeito mais simples possível, um código simples é sempre muito mais fácil de interagir que um mais complexo, portanto sempre que achar uma oportunidade, simplifique ao máximo que puder. Tente evitar de implementar funcionalidades extras no início do projeto, mesmo parecendo que estas serão úteis no futuro, já que na verdade a maioria dessas funções nunca serão usadas no produto final e apenas representarão uma perda de produtividade, bem como o tamanho e complexidade do design do projeto.

Tenha muito cuidado ao tentar reutilizar um design de um projeto antigo. A metodologia XP não aconselha tal prática, já que muitas vezes reutilizar um design significa se utilizar de várias partes consideradas caixas pretas, o que acabará aumentando a complexidade do mesmo.

Posteriormente vem a fase de desenvolvimento e nesta fase a participação do cliente também é uma peça fundamental na elaboração do projeto, será ele quem definirá as estórias, assim como também definirá o que considera mais

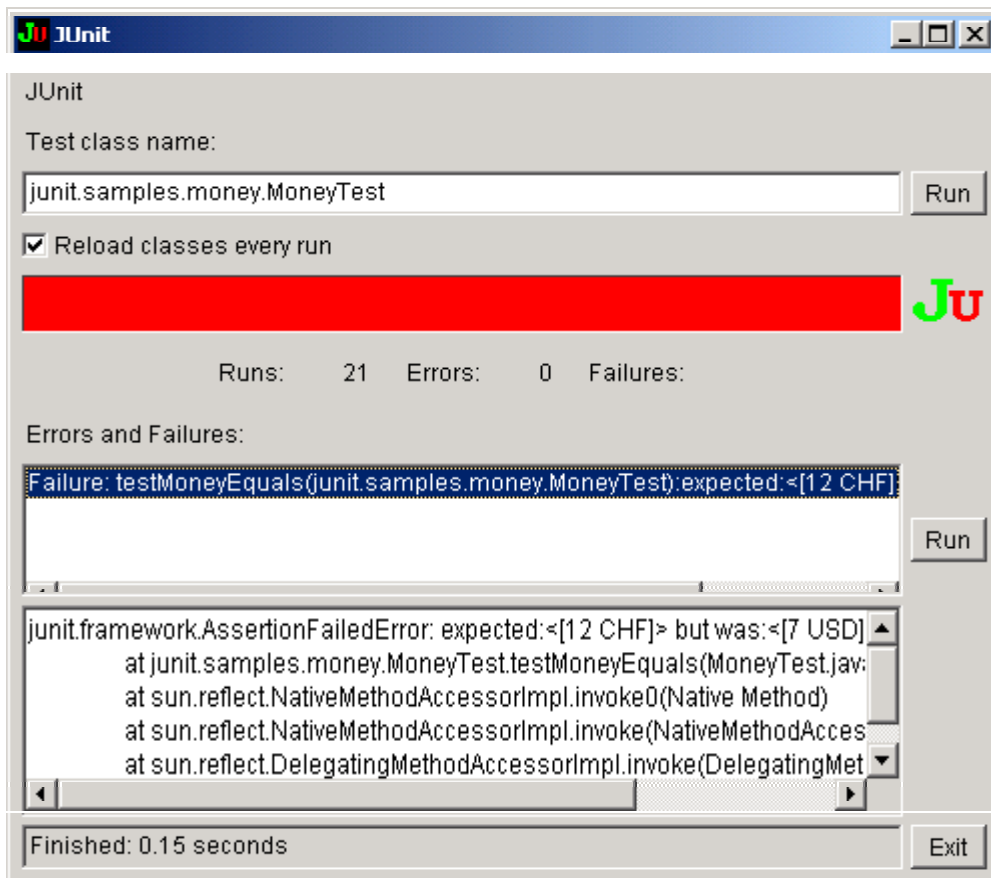
importante para ser colocado em cada nova versão do produto. Com isso em mente, é necessário que este esteja sempre disponível durante a fase de codificação, pois os programadores deverão sempre recorrer a ele em caso de dúvida ou ambiguidade já que a transformação de uma estorieta em uma tarefa nem sempre é trivial (muitos detalhes são omitidos em uma estorieta para atender ao requisito de simplicidade) .

Como foi dito anteriormente, os testes e suas respostas esperadas devem estar definidos antes do início da codificação de qualquer tarefa, deve-se começar a codificação tendo em mente que o código escrito deve ser o mais simples possível para resolver um destes testes, assim que o mencionado teste estiver ok, deve-se realizar o mesmo processo para um próximo teste e assim sucessivamente até que não hajam mais testes a serem realizados. Não implemente nada que não seja estritamente necessário, as funcionalidades adicionais devem ser deixadas de lado já que a maioria delas nunca é usada, gerando apenas mais código e provocando o aumento da complexidade.

O ideal é que todo código seja gerado usando “Pair Programming”, ou seja, duas pessoas programando no mesmo computador. A idéia pode parecer esquisita a primeira vista, e é provável que ocorram vários problemas no início se a equipe não estiver acostumada a trabalhar desta forma. Os benefícios trazidos pelo “Pair Programming” são a geração de um código de mais alto nível e uma melhor distribuição de conhecimentos.

Temos também a fase de testes, e este é um dos pontos chave da metodologia XP, serão os testes que determinarão se uma tarefa está completa ou não, mas para isso é necessário que estes testes tenham a cobertura mais ampla possível, por isso todas as classes da tarefa devem ser testadas, só pode ser deixado de lado o que for extremamente simples.

Quando um “bug” for encontrado e corrigido, deve-se fazer um novo teste que tenha a finalidade de verificar se ele realmente foi resolvido, este teste deve ser automatizado(deve-se fazer uso de uma ferramenta de automatização de testes) juntamente com os demais e repetido sempre que possível [WW].



Nesta figura temos o JUnit, que é um framework que facilita o desenvolvimento e execução de testes em código java. Um dos seus autores é o próprio Kent Beck.

2.2 – OOHDM (Object-Oriented Hypermedia Design Model)

É um método usado para a construção de aplicações hipermídia. É composto de quatro diferentes atividades (projeto conceitual, projeto navegacional, projeto abstrato de interface e implementação) . Durante cada atividade, com exceção da última, alguns modelos orientados a objetos descrevendo o comportamento do projeto são construídos ou aperfeiçoados a partir de iterações anteriores.

Durante o projeto conceitual, um modelo do domínio da aplicação é construído usando-se princípios de modelagem de objetos (OMT), adicionado de algumas primitivas, como perspectivas de atributos e sub-sistemas. Classes conceituais podem ser construídas usando hierarquias. A principal preocupação durante

esta fase é capturar a semântica do domínio com poucos detalhes sobre os tipos de usuários e tarefas. O produto desta atividade é um esquema de classes e instâncias construídos a partir de sub-sistemas, classes e relações [DSS].

É necessário normalmente organizar a informação representada no modelo conceitual, visando ter uma aplicação que possa ser usada por usuários específicos. Em OOHDM, isso é feito definindo-se um modelo navegacional que é uma visão do modelo conceitual. Uma das grandes distinções da uma aplicação hipermídia é a noção da navegação, que deve ser feita sempre tendo-se em mente os diversos tipos de usuários desta aplicação, bem como o conjunto de tarefas que estes devem realizar. Vários modelos navegacionais podem ser feitos para o mesmo modelo conceitual, cada um para um determinado tipo de usuário e tarefa.

O modelo navegacional é composto de dois esquemas, o esquema navegacional de classe e o esquema navegacional de contexto. Os objetos de navegação de uma aplicação hipermídia são definidos pelo esquema navegacional de classe, cujas classes refletem a visão escolhida do domínio da aplicação. Em OOHDM existem uma série de classes de navegação pré-definidas, como nós, links e estruturas de acesso[DSW].

A principal primitiva de um esquema da estrutura navegacional é a noção de contexto navegacional. Um contexto navegacional é um conjunto de nós, links, classes de contexto e outros contextos de navegação. Eles são tirados a partir das classes de navegação, e podem ser definidos através da definição de uma propriedade que todos os nós e links do contexto possuem, ou enumerando seus membros. Por exemplo, um link 1 para n induz um contexto navegacional que permite a ida sequencial para todos os destinos do link (como todas as referências para um conjunto de figuras).

Classes de contexto complementam a definição de classes navegacionais, indicando que informação é mostrada e que âncoras são disponibilizadas ao se acessar um determinado objeto e um contexto particular.

Uma vez que a estrutura navegacional foi definida, ela deve ser feita de forma que seja perceptiva para o usuário através da interface da aplicação, isto é feito

através da fase de projeto abstrato de interface. Essa fase define quais objetos de interface o usuário vai ver, e em particular o modo como diferentes objetos de navegação vão se parecer, quais objetos de interface vão ativar a navegação, como os objetos de interface multimídia vão ser sincronizados e que transformações de interface vão ocorrer.

Em OOHDM é usado a visão abstrata de dados para a descrição da interface do usuário de uma aplicação hipermídia. Visões abstratas de dados são modelos formais orientados a objetos de objetos de interface e são especificados pelo modo que estão estruturados, o modo como estão estaticamente relacionados com os objetos de navegação, e como eles se comportam ao reagirem a eventos externos[OO].

A última etapa é a implementação, e para isso é necessário que se mapeie os modelos navegacionais e abstratos em objetos concretos que estejam disponíveis no ambiente de implementação[DSS]. O modelo gerado depois de se fazer as atividades previamente definidas pode ser implementado de um modo bastante rápido usando-se as plataformas de hipermídia disponíveis, como Hypercards, Toolbook, Director, HTML, etc.

2.3 – RUP (Rational Unified Process)

RUP é um processo de engenharia de software que provê um visão disciplinada para tarefas e responsabilidades de uma organização de desenvolvimento. Tem como meta garantir a produção de softwares de alta qualidade e que atendam as necessidades dos seus usuários finais com um tempo de desenvolvimento e custo previsíveis[PK].

Uma palavra chave desta metodologia é a iteração. O processo iterativo é organizado em fases, essas fases são concepção, elaboração, construção e transição. Essas quatro fases constituem um ciclo de desenvolvimento, no ciclo inicial é criado um software e os ciclos seguintes são os ciclos evolutivos deste software. Cada fase tem uma duração específica que é dependente do projeto, mas para um projeto típico de 2 anos, gastaríamos 2,5 meses na fase de

concepção, 7 meses na elaboração, 12 meses na construção e 2,5 meses na transição. Em cada fase, o progresso é iterativo e consiste de uma ou mais iterações.

Cada iteração contém as atividades de elicitação e análise de requisitos, design e implementação, e integração e testes, mas a ênfase nas atividades muda de acordo com a fase e a iteração.

Na fase de concepção, temos os seguintes objetivo:

- Estabelecer o escopo do projeto, bem como seus critérios de aprovação e o que deve ou não deve estar no produto.
- Tomar nota dos casos de uso mais críticos para o sistema.
- Estimar o custo total do projeto e fazer um cronograma.
- Estimar riscos
- Avaliar os possíveis designs e suas possíveis conseqüências no custo e no cronograma.

Essa fase deve gerar um documento de visão, um modelo de casos de uso, um glossário inicial, uma estimativa de riscos inicial e um plano de projeto(que contém as fases e iterações).

A segunda fase, também conhecida como fase de elaboração, tem como objetivo a análise do domínio do problema, o estabelecimento de uma arquitetura sólida, o desenvolvimento do plano do projeto e a eliminação dos maiores riscos ao projeto. Para atingir esses objetivos é necessário uma visão bastante profunda do sistema.

É necessário portanto que durante essa fase seja construída uma sólida visão dos pontos mais críticos da arquitetura do projeto. Devem ser elaborados a infra-estrutura e o ambiente de desenvolvimento do projeto.

A arquitetura deve ser elaborada e seus componentes selecionados. Deve-se tomar as decisões sempre com consciência e compreensão destes componentes, para que seja possível determinar o custo da fase de construção e seu cronograma com precisão.

Essa fase deve gerar um modelo de casos de usos mais completo que a primeira fase, um documento com os requisitos não funcionais e qualquer requisito que não esteja associado com um caso de uso específico, uma descrição da arquitetura do software, um protótipo da arquitetura, uma lista revisada dos riscos e um plano de desenvolvimento para todo o projeto.

A fase seguinte é a fase de construção, é nela que todos os componentes remanescentes e as características da aplicação são desenvolvidos e integrados no produto, e tudo é testado do começo ao fim [PK]. Essa fase pode ser vista como um processo de manufatura em que a ênfase é colocada no gerenciamento de recursos e controle de operação visando a otimização dos custos, prazos e qualidade.

As atividades essenciais desta fase são as seguintes:

- Gerenciamento e controle de recursos, bem como otimização do processo.
- Desenvolvimento completo dos componentes e fase de testes.
- Lançamento do produto no tempo estimado.

O resultado desta fase é um produto pronto para ir para o usuário final, e deve consistir pelo menos de:

- Um software integrado na plataforma adequada.
- Um manual do usuário.
- Uma descrição da versão atual.

A última fase é a fase de transição, e esta tem como objetivo enviar o produto para a comunidade de usuários e lidar com a resposta desta [IR]. Assim que o produto chegar as mãos da comunidade, esta demandará o desenvolvimento de novas versões, correção de problemas ou a adição de características que não haviam sido implementadas ainda.

Essa fase se inicia quando o software está maduro o suficiente para ser usado pelo usuário final, o que normalmente significa que uma parte do sistema foi

completa dentro de um padrão de qualidade aceitável e sua documentação está disponível para o usuário.

A fase de transição chega ao fim quando o software final atinge o padrão esperado. Para alguns projetos, este final coincide com o começo do próximo ciclo de desenvolvimento, que irá gerar uma nova geração do produto. Para outros projetos, este ponto pode coincidir com a entrega dos artefatos para uma terceira parte que será responsável pela operação, manutenção e melhoria do sistema entregue [PK].

3 - Vantagens e desvantagens individuais

Esse capítulo se dispõe a apresentar algumas das vantagens e desvantagens mais significativas de cada uma das metodologias de desenvolvimento citadas. A análise será puramente individual, uma vez que ainda não estabelecemos critérios para fazer uma comparação entre elas.

3.1 - XP

Vantagens:

- Se adapta bem a sistemas com mudanças constantes de requisitos.
- Ideal para equipes de desenvolvimento não muito grandes (menos de 10 pessoas).
- Excelente para se trabalhar com prazos de desenvolvimento apertados.
- É uma metodologia leve e não gera uma infinidade de documentos que não serão usados depois.
- Alta produtividade.
- Testes automatizados e contínuos facilitam em muito a descobertas de erros para sua posterior correção.
- É uma metodologia extremamente flexível.

- Pode ser aprendida em pouco tempo.

Desvantagens:

- Não é ideal se usar quando a equipe é composta de muitos integrantes(mais de 15).
- O trabalho pode não ser possível caso a equipe não tenha o perfil necessário para se usar XP. Podemos citar por exemplo, profissionais que não gostem de trabalhar em equipe.
- Instalações físicas apertadas podem impossibilitar o uso de XP.
- Quando o sistema tem um tempo de resposta alto, os testes automatizados podem ser inviabilizados.

3.2 - OOHDM

Vantagens:

- É uma metodologia exclusivamente voltada para a construção de aplicações hipermídia, fazendo muito bem o que se propõe neste escopo.
- Apesar de propor uma notação (muito semelhante a UML), OOHDM não faz imposições, a ênfase está mesmo no caminho que se sugere para a construção de aplicações hipermídia.
- O tempo de aprendizado é bem curto.
- A metodologia é extremamente simples e leve.

Desvantagens:

- Serve apenas para a construção de aplicações hipermídias.
- Possui pouco material disponível se comparado as outras metodologias, além de não possuir um livro publicado que aborde o assunto em mais detalhes.

3.3 - RUP

Vantagens:

- Controla continuamente a qualidade do software.
- Possui integração com a suíte da Rational.
- A metodologia está sempre em evolução.
- É usado por um grande número de grandes companhias (Ericsson, Xerox, Visa, etc).
- Cada fase é bem definida e possui diversos templates.
- O fato da identificação dos riscos ser feita no começo, diminui o risco de atrasos no projeto.
- Suporta bem a mudança de requisitos durante o projeto.
- Se acomoda bem a diferentes tipos de projetos (tamanho, tipos, competência, etc).

Desvantagens

- É um processo muito rígido.
- Necessita de um tempo maior para o aprendizado.
- Gera um grande número de documentos.

4 - Estabelecimento dos critérios de comparação.

Este capítulo se propõe a estabelecer quais critérios serão usados para se efetuar a comparação entre as três metodologias citadas. Além do estabelecimento destes critérios, será feita uma breve descrição de o que cada um representa.

1. Preço de implantação

Esse critério tem como base verificar o custo para a implantação da metodologia especificada em uma empresa. Será verificado se a metodologia

possui algum custo, seja em manuais, softwares ou hardware. Esse item é de extrema importância, já que a adoção de uma metodologia em uma determinada empresa (principalmente pequenas e médias) pode ser inviabilizada devido a falta de recursos necessário para sua implantação.

2. Custo de treinamento

Deve-se ter em mente que para que a implantação de uma metodologia seja bem sucedida em uma empresa, não é necessário apenas ter os manuais ou softwares necessários para o seu uso, é necessário também que seja feito um treinamento da equipe envolvida, já que assume-se que nenhum dos seus integrantes teve contato prévio com a metodologia em questão. Será avaliado o custo envolvido neste treinamento e se é necessário que ele seja formal ou se é possível que a metodologia seja aprendida apenas através de livros ou materiais disponíveis.

3. Tempo para o aprendizado

Além do custo de treinamento, deve ser levantado também o tempo que é necessário para que seja realizado o aprendizado da metodologia. Isso é especialmente importante quando se quer fazer uso de uma determinada metodologia sem prévio conhecimento, em um projeto com um prazo bem definido. Dependendo da metodologia, o prazo do projeto pode se esgotar antes mesmo que o aprendizado desta tenha terminado.

Cada metodologia possui uma curva de aprendizado diferente, e muitas vezes elas são radicalmente diferentes, deve-se ter isso em mente ao se decidir por uma metodologia específica.

4. Tamanho necessário da equipe

Cada projeto possui um número específico de integrantes na equipe, e metodologias diferentes se adequam de forma diferente para um determinado tamanho de equipe. Deve-se considerar esse fato ao se fazer a escolha da metodologia, já que a produtividade pode ser sensivelmente afetada caso esta não se adapte ao tamanho da equipe do projeto.

5. Perfil necessário

Esse item é muito pouco considerado ao se optar por uma metodologia, mas ele não deve ser subestimado de forma alguma, já que pode significar a diferença entre o sucesso ou não da implantação.

Certas metodologias necessitam que seus praticantes tenham um determinado perfil, seja este o perfil de uma pessoa burocrática ou de uma que consiga trabalhar bem em equipe. Caso esse perfil demandado seja ignorado, os resultados podem ser desastrosos e isso pode acabar por inviabilizar o sucesso do projeto como um todo.

6. Nível profissional necessário da equipe

Esse é outro item que não costuma ser muito estudado no processo da escolha de metodologia a ser adotada. Deve-se levar em conta sempre o nível profissional da equipe envolvida e o nível exigido pela metodologia, um exemplo disso seria adotar uma metodologia que necessite de profissionais com pós-graduação em uma equipe que só possui estagiários.

Em alguns casos, o nível profissional exigido por uma metodologia é tão grande que o custo com a folha de pagamento dos integrantes da equipe inviabiliza o projeto.

7. Documentação disponível

É importante ter em mente a disponibilidade de documentação antes de tomar a decisão final sobre qual metodologia escolher. Ter material para consulta e estudo é essencial para o sucesso da implantação desta no ambiente de trabalho, quando mais material disponível melhor. Esse material pode ser em forma de livros, teses, textos na web, etc.

8. Ferramentas disponíveis

Além da documentação, a disponibilidade de ferramentas de apoio também é outro fator importante de qualquer metodologia, já que as ferramentas podem automatizar certos processos repetitivos, ajudar no controle de prazos e tantas outras tarefas que demandariam tempo e esforço por parte da equipe.

9. Possibilidade de reuso de código

Muitas empresas se utilizam largamente do reuso de código em seus projetos, deve-se portanto saber se a política da metodologia em relação ao reuso, ou seja se aconselha ou mesmo ajuda no reuso de código.

10. Possibilidade de conflitos

A inclusão de uma metodologia de desenvolvimento em um local onde antes não havia nada, pode vir a ser pouco receptiva. Essa inclusão pode vir a gerar conflito entre os membros da equipe, sendo que isso é especialmente preocupante caso a introdução da metodologia resulte em mudanças severas no modo de como o desenvolvimento sempre foi feito na empresa.

Essa falta de receptividade pode vir a comprometer por completo a implantação da mesma, já que ela pode vir a resultar em conflitos internos com os membros que relutam em sua adoção.

11. Benefícios para a empresa.

Toda empresa objetiva algum benefício ao adotar uma metodologia de desenvolvimento, cabe saber se a metodologia escolhida propõe trazer tais benefícios. Esses benefícios variam muito de empresa para empresa, podendo ser desde um aumento na produtividade até mesmo a obtenção de algum certificado de qualidade.

5 - Comparação entre as metodologias

Esse capítulo tem como base fazer finalmente a comparação entre as metodologias que foi proposta no início deste documento.

Vale ressaltar que o autor em questão não possui qualquer vínculo com nenhuma das metodologias citadas, e que a comparação tentará ser a mais imparcial possível.

A comparação será feita com base nos critérios definidos no capítulo anterior.

Preço de implantação:

XP e OOHDM são metodologias com um preço de implantação muito pequeno, já que muito de suas documentações podem ser conseguidas gratuitamente na internet. Não é necessário nenhum hardware específico para fazer uso de qualquer uma delas, assim como não é necessário nenhum manual ou livro específico, embora o investimento em um livro de XP seja recomendado, já que existem alguns bons exemplares disponíveis (Kent Beck – Extreme Programming Explained e William C. Wake – Extreme Programming Explored são dois bons exemplos).

Existe uma ressalva em relação a XP, já que como já foi explicado, é necessário que a empresa possua um ambiente propício a interação, ou seja, as pessoas que irão interagir devem estar fisicamente perto uma da outra, trabalhar em mesas que permitam a programação em pares, etc. Caso a empresa que deseje adotar a metodologia não possua as instalações necessárias, o custo para sua implementação irá incluir uma reforma no ambiente, o que pode inviabilizar o orçamento.

Quanto a RUP, dependendo de como faremos a implantação, podemos ter um custo tanto elevado quando bem baixo. A Rational aconselha que se use a sua suíte para se utilizar RUP com máxima eficiência. Essa suíte possui diversas ferramentas que irão ajudar em muito o uso eficiente da metodologia, mas fará com que o preço final possa ficar inviável para uma empresa que esteja em seus estágios iniciais. Por outro lado, não é necessário ter essa suíte para a implantação, e até mesmo a documentação pode ser conseguida gratuitamente (apesar de existirem alguns bons livros disponíveis). Devido a complexidade da metodologia, é aconselhável se comprar a suíte da Rational (várias suítes são disponíveis e o preço varia entre \$5,000.00 e \$10,000.00) ou o RUP®, best practices (que é mais barato que a suíte e custa por volta de \$700.00), mas novamente isso pode inviabilizar o orçamento de implantação para empresas pequenas. Para rodar a suíte é necessário ter um hardware um pouco acima do básico.

Custo de treinamento

Novamente XP e OOHDM tem um custo de treinamento bem baixo. Ambas metodologias são de simples aprendizado e não é necessário um treinamento mais formal por parte da equipe do projeto, todo aprendizado pode ser feito através de livros (no caso de XP, já que OOHDM ainda não possui nenhum livro publicado) e documentos que podem ser achados gratuitamente na internet. O custo dos livros de XP não são elevados e estes são boas fontes de conhecimento sobre a metodologia.

Uma boa opção é ter alguém já familiarizado com a metodologia na equipe, essa pessoa pode passar esse conhecimento para os demais sem um custo adicional para a empresa.

No caso de RUP, devido a natureza mais complexa da metodologia, aconselha-se a fazer um curso mais formal, e a própria Rational oferece diversos destes cursos. Existe abundante material gratuito na internet sobre ela, mas é aconselhável ter um livro de referência (como o The Rational Unified Process, An Introduction - Philippe Kruchten).

Novamente, não é necessário se fazer um curso de RUP ou mesmo comprar qualquer livro relacionado(a equipe usaria apenas a documentação conseguida gratuitamente na internet), mas essa prática não é aconselhável no caso do RUP devido a complexidade da metodologia.

Tempo para o aprendizado

Tanto OOADM quando XP são metodologias ditas “leves”(metodologias com poucas regras e práticas, todas fáceis de se seguir) e portanto não possuem uma curva de aprendizado muito alta.

OOADM além de ser uma metodologia “leve”, possui um domínio mais restrito (apenas aplicações hipermídias) e portanto é bem simples de se aprender. Caso o usuário seja conhecedor de princípios de modelagens de objetos, o aprendizado será ainda mais rápido, já que maior parte do tempo gasto para aprender essa metodologia é usado no aprendizado de orientação a objetos e seus princípios de modelagem (OMT).

XP também é uma metodologia “leve”, mas pelo fato de possuir um domínio mais amplo, tem uma curva de aprendizado ligeiramente maior que OOADM. É possível aprender todas as suas regras e práticas em um curto período de tempo (é possível aprender tudo em poucos dias, mas vai demorar mais tempo para se familiarizar com todas elas).

RUP é uma metodologia mais “pesada” e portanto possui um número bem maior de regras e práticas a serem seguidas e portanto possui uma curva de

aprendizado consideravelmente maior do que as duas metodologias anteriores. É necessário aprender a fazer corretamente cada documento gerado por cada uma de suas fases, e isso pode vir a levar algum tempo(são necessários alguns meses para se ficar familiar com todas as regras e práticas de RUP).

Tamanho necessário da equipe

XP é uma metodologia que reconhecidamente funciona muito melhor com equipes pequenas(10 ou menos). Isso se deve ao fato de que como é necessário que haja sempre uma grande interação entre todos os membros da equipe, quando maior esta, maior será o tempo gasto com essa comunicação, logo caso a equipe seja muito grande será gasto mais esforço com a interação do que efetivamente com a produção. Existem alguns casos documentados de experiências de sucesso com projetos usando XP com equipes muito grandes (por volta de 50 membros), mas estes casos são raros. Existe também uma experiência em andamento, onde se tem por objetivo dividir uma equipe muito grande em vários grupos pequenos e usar XP em cada um deles, mas não existem dados conclusivos sobre isto ainda.

RUP ao contrário de XP, funciona melhor com equipes maiores, isso se deve ao fato de que a quantidade de documentos gerado em cada fase é enorme, e portanto é necessário ter pessoal para que estes documentos sejam produzidos a contento. Esta metodologia já foi testado várias vezes com projetos de grande porte e sempre apresentou bons resultados quando bem aplicada(seus passos foram seguidos a contento). É dito que XP não passa de uma instância de RUP voltada para projetos menores, e para provar isso existem alguns variantes de RUP voltados para esses projetos menores, um dele é o dX [OOA] que propõe uma instância menor de RUP com algumas técnicas de XP.

OOHDM não especifica qual o tamanho ideal para a sua equipe, mas existem casos de sucesso tanto com equipes pequenas, média e grandes.

Perfil necessário

XP necessita de uma equipe de membros com um perfil comunicativo e que gostem de trabalhar em equipe. Esse perfil é absolutamente essencial para o sucesso do XP, a interação é a palavra chave para essa metodologia, caso os membros da equipe não gostem de trabalhar em equipe e não sejam comunicativos, o projeto ficará inviável. É necessário portanto ter uma noção do perfil da equipe antes de optar por XP, pois caso a equipe não tenha o perfil necessário, o projeto poderá acabar em desastre.

RUP não necessita tanto de um perfil específico como XP, mas devido a quantidade de regras a serem seguidas e documentos que devem ser gerados, é bom que a equipe tenha um perfil de pessoas mais organizadas e metódicas, para que todos esses passos sejam seguidos a contento.

OOHDM também não necessita de um perfil específico para sua implantação. Não é necessário ser tão metódico quanto RUP, mas é necessário ter perfil para seguir as fases demandadas pela metodologia.

Nível profissional necessário da equipe

Tanto XP quando OODHM são metodologias bem simples, e não é necessário um grau profissional muito alto para seguir suas regras, valores e princípios. É altamente recomendável porém que o profissional possua uma boa noção de orientação a objetos e no caso de OOHDM, possua algum conhecimento de OMT. Fora esses pré-requisitos, não é necessário nem que o profissional possua nível superior completo(embora isso seja desejável).

No caso de RUP, é necessário um nível profissional maior do que as outras duas metodologias analisadas. Isso se deve ao fato desta ser uma metodologia mais “pesada” e portanto requer um formalismo maior por parte dos profissionais envolvidos. É necessário saber não apenas o que cada fase gera, mas também como gerar cada um dos documentos pedidos, o que não é tarefa trivial devido

ao grande número que é pedido. Aconselha-se que os profissionais responsáveis tenham pelo menos nível superior.

Documentação disponível

XP E RUP possuem ampla bibliografia disponível, tanto na forma gratuita quanto paga.

XP é uma metodologia que está em “moda” hoje em dia e portanto existem diversas páginas na internet com ampla documentação, focados em diversos domínios. A documentação é tão vasta que até elimina a obrigatoriedade de se comprar um livro específico do assunto como referência. Existem diversos papers, teses e livros sobre o assunto, dessa forma no quesito documentação, a empresa que optar por XP estará bem servida.

RUP por sua vez é uma metodologia mais antiga que XP e já foi bastante testada pelo mercado. Como resultado disto, existem diversos documentos sobre o assunto, sejam estudos de casos, variações para a metodologia, relatos, descrições, etc. É possível comprar na Rational o RUP®, best practices que é um conjunto de ferramentas de acompanhamento web para RUP. Existem também alguns livros publicados, bem como uma infinidade de páginas na rede sobre o assunto.

No caso do OOHDM, a documentação disponível é provavelmente um de seus pontos mais fracos. Não existe livro publicado sobre o assunto e a documentação que pode ser obtida se encontra ou em algumas teses (de alcance restrito para alunos) ou algumas páginas na Internet. Apesar de existirem algumas boas páginas, ainda falta uma documentação mais completa da metodologia.

Ferramentas disponíveis

A metodologia melhor servida de ferramentas sem dúvida é RUP. Isso se deve ao fato de que a metodologia foi desenvolvida pela equipe da Rational e esta

vende softwares específicos para ela. É possível comprar um pacote que possui entre outras coisas o Framework do RUP, um modo de visualização Web de todas os processos e tarefas, e a possibilidade de instalar Plug ins obtidos no site da Rational. Além desse pacote, é possível comprar a suíte de ferramentas da Rational, que pode ajudar em muito no andamento do projeto.

No caso de OOHDH e XP, ambas não possuem ferramentas específicas como é o caso de RUP, mas é possível usar outras ferramentas que estão disponíveis no mercado. É possível usar ferramentas como o Project da Microsoft, apesar de que no caso de XP não existe um conceito de fluxo incremental e recursivo de projetos e atividades, por isso muitos usam ferramentas ainda mais simples para gerenciar o projeto.

Para OOHDH, usa-se ferramentas genéricas para modelagem e controle de fluxo. Pode-se fazer uso até mesmo de um editor de texto com capacidade gráfica como o Word da Microsoft para a modelagem, e um produto como o já citado Project para o controle de fluxo.

Possibilidade de reuso de código

OOHDH possui uma ênfase muito grande no reuso em todos os 4 passos do processo. Classificação, agregação e generalização/especialização são usados durante todo o processo para maximizar a abstração e as oportunidades de reuso [TEL].

RUP prega que pode-se enxergar componentes reusáveis como produtos de softwares, assim eles devem ter um bom design, ser testados, documentados, etc. Dessa forma um projeto pode produzir componentes reusáveis para outros projetos. O foco que RUP dá para a arquitetura é a fundação para o reuso, existem na metodologia muitas instruções de como se usar frameworks de arquitetura e patterns, e como incorporar elementos já existentes em um projeto [RA].

Ao contrário das duas metodologias anteriores, XP não recomenda que se codifique pensando em reuso a não ser que se esteja certo de que o reuso será

necessário. Isso vai contra uma das regras de ouro da programação orientada a objetos, mas é um reflexo de como as coisas funcionam no mundo real e vai de encontro com a filosofia do XP de simplicidade (faça sempre do modo mais simples possível que funcione e não refaça o código para possíveis futuros requisitos) [XPOO].

Possibilidade de conflitos

Caso não haja anteriormente na empresa nenhuma metodologia implantada, a introdução de uma metodologia dita “leve” é recomendada, já que não implica em uma grande mudança no ambiente da empresa em questão. XP e OOHDM portanto são uma boa opção para empresas que estão querendo adotar uma metodologia de desenvolvimento, mas nunca tiveram experiência com outras.

RUP por sua vez é mais “pesada” que as duas anteriores, e portanto deve-se ponderar muito antes de adotá-la caso os membros da equipe não possuam contato anterior com outras metodologias. O ideal é portanto adotar RUP apenas quando a equipe já possuir pelo menos um nível básico de conhecimento com metodologias de desenvolvimento.

Benefícios para a empresa

XP é uma metodologia “leve” que agrega para a empresa um conjunto de regras, valores e princípios que são simples e permitem o rápido desenvolvimento de aplicações sem comprometer sua qualidade e satisfazendo plenamente os requisitos passados pelo cliente.

É uma metodologia com um custo de implantação e treinamento baixo, assim como uma curva de aprendizado pequena e ampla documentação disponível. Está muito em voga nos dias de hoje e tem portanto um bom número de seguidores. Tem eficiência comprovada para projetos com equipes que não ultrapassem 10 pessoas, pode ser usada também para projetos maiores, mas recomenda-se cuidado ao se fazer.

Tem excelente adaptação à mudanças de requisitos e é ideal para se trabalhar com prazos apertados já que não gera muitos documentos e tem comprovadamente alta produtividade.

OOHDM também é uma metodologia simples, mas essa possui um escopo mais limitado que as duas outras analisadas neste documento e tem as vantagens e desvantagens que esse escopo propõe ou seja, a metodologia serve apenas para o desenvolvimento de aplicações hipermídias, mas em compensação o faz melhor que as duas anteriores, portanto se a aplicação que se visa desenvolver for hipermídia, OOHDM é uma excelente opção.

É uma metodologia de fácil aprendizado e baixo custo de implantação, mas peca na parte relativa a documentação. A maioria da documentação que pode ser obtida sobre OOHDM está relacionada com o mundo acadêmico e não existe um livro publicado sobre o assunto, a melhor opção é portanto obter essa documentação pela internet.

RUP por sua vez é uma metodologia bem mais pesada que as anteriores e possui um grande conjunto de regras, valores e princípios a serem seguidos, mas em compensação é completamente integrado as ferramentas da Rational, o que facilita muito o processo.

A curva de aprendizado assim como o nível profissional necessário não são pequenos, mas a documentação a respeito da metodologia é ampla, existindo vários livros e artigos publicados sobre o assunto.

RUP trás um formalismo mais forte para a empresa, o que pode ou não ser uma vantagem. Essa metodologia é ideal para projetos maiores, onde essa documentação vai ser realmente necessária e onde outras metodologias (como XP) são falhas. Existem algumas instâncias de RUP que são exclusivas para projeto menores, diz-se inclusive que XP não passa de uma dessas instâncias.

Bibliografia

[KB] Kent Beck – Extreme Programming Explained – 1999

[WW] William C. Wake – Extreme Programming Explored – 2000

[GI] Extreme Programming a Gentle Introduction –

<http://www.extremeprogramming.org/>

[XR] XP programming.com na Extreme Programming Resource –

<http://www.xprogramming.com/>

[DSW] Daniel Schwabe - OOHDM- WEB: Rapid prototyping of hypermedia applications in the web.

[DSS] Daniel Schwabe - Systematic Hypermedia application design with OOHDM.

[OO] OOHDM <http://www-lifia.info.unlp.edu.ar/~fer/oohdm/>

[PK] Philippe Kruchten - The Rational Unified Process, An Introduction

[RA] Rational – www.rational.com

[IR] Introducing the RUP into an Organization

http://www.therationaledge.com/content/jan_02/m_introducingTheRUP_dd.html

[TEL] Telemídia – Puc-Rio - <http://www.telemidia.puc-rio.br/oohdm/oohdm.html>

[OOA] Grady Booch, Robert C, Martin, James Newkirk - Object Oriented Analysis and Design with Applications, 2d. ed. 1998

[XPOO] XP vs OOP -

<http://archive.devx.com/free/articles/2000/Abbott01/Abbott01-4.asp>