Um Esboço do Processo de Desenvolvimento

Professor: Gilmar Luiz de Borba

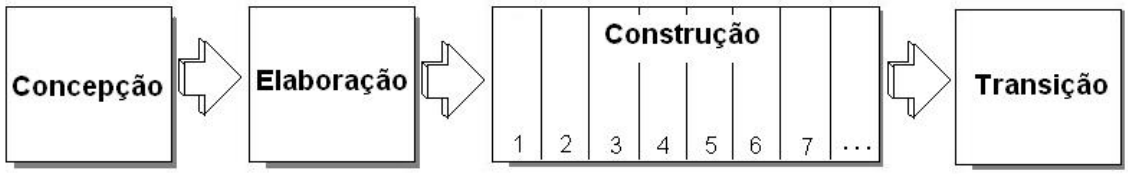
1.0 - RESUMO

Segundo Fowller, A UML não possui noções dos processos, mas para ele, a técnica de modelagem não tem sentido sem que se saiba como elas se encaixam nos processos de desenvolvimento. Para o melhor entendimento do desenvolvimento orientado a objetos, baseado na UML, será discutido nessa resenha, as fases do processo unificado da Rational, (RUP) e a sua contextualização na linguagem UML conforme detalhado no capítulo 2 do livro de Fowler e Scott.

2.0 - INTRODUÇÃO

Os três amigos (Ivar Jacobson, Grady Booch e James Rumbaugh) além de desenvolverem a UML (Unified Modeling Language), desenvolveram também um processo unificado, denominado Rational Unified Process (antigo Objectory). Não será discutido neste artigo o Rational Unified Process, porém, serão discutidos os processos baseados na visão dos três amigos. Qualquer tipo de processo pode ser tratado com a UML. A UML é independente de processo. No desenvolvimento de software podemos ter vários tipos de processo, tais como: em tempo real, sistemas de informações e processo contidos em produtos de desktop.

3.0 – O PROCESSO DE DESENVOLVIMENTO – VISÃO GERAL



A figura acima mostra uma visão geral (de alto nível) das fases do processo de desenvolvimento RUP. Foi dado ênfase para as iterações na fase de construção, pois esta fase é a fase-chave das interações, porém, o que acontece na realidade são iterações também nas demais fases.

3.1 - CONCEPÇÃO:

Trata-se da idéia inicial do projeto, exemplo: “Vamos colocar nosso catálogo de produtos na WEB” ou “Vamos implementar nosso sistema de atendimento ao cliente em todo território nacional”. Como podemos ver, a concepção pode ser um “bate-papo” de meia

hora ou levar meses para se consolidar, necessitando inclusive de estudos de viabilidade complexos. Nesta etapa é elaborado o plano de negócio: quanto custará, quanto tempo levará, qual será a dimensão do projeto. Essa etapa será determinante para que o patrocinador concorde em continuar a discussão indo para a fase de elaboração.

3.2 - ELABORAÇÃO:

Essa etapa fornecerá condições para identificar o que realmente será construído e como será construído. Os riscos do projeto devem ser levados em consideração com mais seriedade, nessa fase. Os riscos podem ser, por exemplo: de requisitos, tecnológicos, habilidades e políticos. O ponto inicial de um processo de desenvolvimento orientado a objeto é a utilização do diagrama de **CASO DE USO**. O diagrama de Caso de Uso é a fermenta de comunicação entre clientes e desenvolvedores, é por esta razão que nesta fase são agendadas entrevistas com os usuários, justamente para identificar os casos de uso. Aliado ao diagrama de Caso de Uso podemos também usar os diagramas de **CLASSES**. Os diagramas de Classes são usados para descrever o modelo de domínio[1]. Este diagrama normalmente é usado para sintetizar os conceitos que os analistas de negócio possuem do sistema. Se o domínio possuir um forte elemento de WorkFlow também pode ser usado o diagrama de **ATIVIDADES**. Esses diagramas ajudam na identificação de processos em paralelo, eliminando processos seqüenciais desnecessários. Outro diagrama da UML que pode ser usado nessa fase é o diagrama de **INTERAÇÃO**, este diagrama identifica como vários papéis interagem em um negócio, ou seja, como grupos de objetos colaboram em algum comportamento, são usados para capturar o comportamento de um único Caso de Uso, detalhando-o.

3.3 - FASE DE CONSTRUÇÃO

Antes de iniciar a fase de construção é necessário planejá-la. O ponto crítico do planejamento é determinar a dimensão de cada iteração (processo), a quantidade de recursos técnicos e o tempo necessário para executá-la. A construção deverá ser feita em uma série de iterações, segundo FOWLER, SCOTT, (2000), “cada iteração é um miniprojeto”. Na UML, usa-se normalmente o diagrama de **CASO DE USO** como base para este dimensionamento e é por este motivo que a UML dá muita ênfase a esse

[1] Modelo de Domínio: descreve um modelo onde o sujeito primário é o universo que o sistema de computação suportará em qualquer estado do processo de desenvolvimento.

diagrama. O cliente (analista de negócio) define o grau de prioridade para cada **CASO DE USO**, exemplo: alta, média e baixa, o cliente também descreve o conteúdo de cada uma dessas prioridades de acordo com as suas regras. Por outro lado, o desenvolvedor, divide os **CASOS DE USO** de acordo com o risco: alto risco, médio risco e baixo risco, de acordo com o seu know-how de desenvolvimento. Baseado nestas estimativas e em função da quantidade de desenvolvedores, tempo previsto para um processo e um coeficiente de segurança (baseado no tempo ideal e real de desenvolvimento), pode ser determinado uma primeira estimativa para execução de uma tarefa (um **CASO DE USO**) e conseqüentemente do projeto.

A construção consiste em uma série de atividades para cada **CASO DE USO**, são elas: análise, projeto, codificação, teste de unidade, teste de integração e documentação. Ao final de todas estas atividades o desenvolvedor submete a tarefa ao usuário final (teste de aceitação) de modo a confirmar que o **CASO DE USO** em questão foi definido e construído corretamente.

Nessa fase podem ser usados ainda os diagramas de **PACOTES**, como um mapa lógico, mostrando as dependências e auxiliando o desenvolvedor a mantê-las sob controle. O diagrama de **UTILIZAÇÃO**, também é útil nessa fase, ele ajuda a mostrar um quadro físico do sistema. Dentro de cada pacote pode ser apresentado o diagrama de **CLASSES** de modo a mostrar uma perspectiva de associações e atributos-chave. Se determinada classe resultará em objetos com um comportamento de vida complexo, com mudanças de estados abundantes é aconselhável o desenho do diagrama de **ESTADOS**. Se há um algoritmo complexo poderá ser usado o diagrama de **ATIVIDADE** para demonstrar essa lógica complexa.

3.4 – TRANSIÇÃO

Algumas atividades não devem ser feitas em etapas iniciais, uma delas é a otimização. A otimização precoce faz com que o desenvolvimento se torne mais difícil, reduz a claridade e capacidade de extensão do sistema. Por este motivo esta atividade é realizada normalmente nesta etapa final do desenvolvimento. Esta fase é ideal também para corrigir erros (bugs) restantes e fazer a passagem da versão BETA para a versão final do sistema. O uso de padrões de projeto será útil para possibilitar uma linguagem única entre a equipe de desenvolvimento, evitar a redundância de códigos e possibilitar qualidade, uma vez que os padrões já foram testados pela indústria de software. Portanto o uso de

padrões será de grande importância a partir da fase de construção e facilitará a solução de problemas na transição.

4.0 - CONCLUSÃO

Discutimos a partir dessa resenha, que a UML permeia o processo de desenvolvimento, neste caso, usamos as fases do processo unificado da Rational, o RUP, para verificar como isso acontece. Isso poderá ser válido para outros processos de desenvolvimento.

Conforme esclarece o autor, qualquer tipo de processo pode ser tratado com a UML, como também não existe um único processo para o planejamento, implementação e documentação do software, isso vai depender do tipo de software que será produzido, e a equipe técnica pode criar seu próprio processo, esclarece o autor:

> [...] vários fatores associados com o desenvolvimento de software levam a vários tipos de processos. [...] As equipes devem criar seus próprios processos, utilizando processos publicados como orientação e não como padrões a serem seguidos. FOWLER, SCOTT (2000:30)

O texto dá ênfase ao desenvolvimento iterativo na fase de construção, o uso da iteração é essencial, pois pode ser usada para expor de maneira antecipada os erros e os riscos do projeto e obter maior controle do desenvolvimento. Finalmente o texto salienta sobre a importância do uso de padrões, acrescenta-se aqui, que os padrões permitirão o uso compartilhado de um vocabulário; melhoram o processo de comunicação, ou seja, "dizem mais, com menos"; fornecem qualidade e possibilitam uma experiência comprovada do uso da OO.

REFERÊNCIAS

FOWLER, Martin.; SCOTT Kendal. ***UML essencial: um breve guia para a linguagem padrão de modelagem de objetos***. 169 p. 2. ed. – Porto Alegre: Bookman, 2000.