

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RAPID APPLICATION DEVELOPMENT

Disciplina: Modelagem a
Programação Orientada a Objetos

Docente: Giordano Cabral

Discente: Flaviano Dias Fontes

Recife/2012

Sumario

Introdução	1
Metodologia Iterativa	2
Prototipagem de Software.....	3
Desenvolvimento Rápido de Aplicações	5
Conclusão.....	8
Referencias Bibliográficas	9

Introdução

Rapid application development (em português: Desenvolvimento rápido de aplicativos) é uma metodologia de desenvolvimento que visa uma prototipagem rápida com o mínimo de planejamento. O seu planejamento é intercalado com a escrita do código. Por não ter um planejamento prévio permite que o *software* possa ser escrito muito mais rápido e que os requisitos também possam ser alterados de forma muito mais fácil.

Metodologia Iterativa

O Rapid Application Development (RAD) é uma metodologia de desenvolvimento de software envolve em um de seus métodos o desenvolvimento iterativo e incremental. A ideia básica por trás desse método ágil é desenvolver um sistema respeitando um ciclo repetitivo e em pequenos pedaços por vez, dessa forma os desenvolvedores podem tirar proveito do que foi aprendido durante o desenvolvimento de versões anteriores do sistema. Essa aprendizagem vem tanto do processo de desenvolvimento quanto o de utilização do sistema.

Quando possível as principais etapas do processo começam com uma simples aplicação de uma parte dos requisitos de software que frequentemente aprimoram as versões em evolução até que o sistema completo seja implementado. Em cada ciclo, as modificações no projeto são feitas e novas capacidades funcionais são adicionadas.

As fases incrementais da metodologia iterativa são divididas em grupos de processos unificados, essas fases podem ser divididas em uma ou mais iterações que geralmente tem uma duração fixa, são eles:

Criação: Identifica o escopo do projeto, os requisitos (funcionais e não funcionais) e os riscos a um nível de detalhe em que possa ser estimado.

Elaboração: Nessa fase é oferecida uma arquitetura de trabalho para que possa ser reduzido os principais riscos e preenchidos os requisitos não funcionais.

Construção: A partir da análise, do projeto, da implementação, dos testes e dos requisitos funcionais estará pronta a arquitetura para a produção de código.

Transição: É a fase que proporciona o sistema ir para o ambiente de operação.

As diretrizes que orientam a implementação incluem: Revisão da implementação atual para que seja sempre verificado se está atendendo os objetivos do projeto, Opiniões dos usuários devem ser solicitada para que investigue a possível ocorrência de uma deficiência da implementação atual, Modificações devem ser fáceis de serem realizadas se não forem deve existir um problema básico como uma falha de projeto, entre outras.

Prototipagem de Software

Uma das metodologias envolvidas no RAD é a de prototipagem de software que se refere como o próprio nome já diz a criação de protótipos das aplicações que o software final vai ter. A prototipagem de software pode ser comparada à prototipagem que ocorre na indústria automobilista, eletrônica e de produção, pois a concepção de um protótipo visa ter a antecipação dos possíveis problemas e falhas antes de chegar no usuário final.

A prototipagem tem diversas vantagens algumas delas são: O designer de software pode obter um valioso feedback dos usuários logo no início do projeto, o cliente e o contratante podem comparar se o produto mostrado corresponde ao que foi solicitado nos requisitos e o engenheiro de software pode verificar a precisão daquilo que foi estimado inicialmente e verificar se os prazos e metas estipulados poderão ser cumpridos com êxito.

O processo de prototipagem envolve as seguintes fases:

Identificar os requisitos básicos: É a determinação dos requisitos básicos incluindo a entrada e a saída desejada, a segurança da informação geralmente é ignorada.

Desenvolvimento do protótipo inicial: O protótipo inicial é desenvolvido apenas com a interface do usuário, conhecida como protótipo horizontal, que fornece uma visão ampla de todo o sistema, focando com a interação com o usuário, esses protótipos são úteis para confirmação dos requisitos de interface com o escopo do sistema, Versão de demonstração para a venda do sistema e até desenvolver estimativas de custo, tempo e esforço de desenvolvimento.

Análise: Os clientes, incluindo os usuários finais do sistema, examinam o protótipo com o intuito de fornecer um feedback que poderão ser aplicados em alterações no sistema.

Revisão para melhoria do protótipo: Utilizando os feedbacks da análise as especificações e os próprios protótipos podem ser melhorados com as alterações feitas será necessário realizar uma nova análise para verificação da eficácia das melhorias.

A prototipagem vertical é uma elaboração mais robusta de um método e sua utilidade é para a obtenção de requisitos mais detalhados, como esses: Projeto de banco de dados mais refinado, obtenção de informações sobre um volume de dados e as necessidades de interface do sistema, como também para esclarecimento de exigências muitas vezes complexas na implementação do sistema.

Os dois tipos principais de prototipagem são a Prototipagem Throwaway e a Prototipagem Evolutiva.

A prototipagem Throwaway é uma prototipagem rápida descartável pois é criado um modelo que será descartado e não irá compor o produto final, sua utilização é porque pode ser feita de forma rápida o que fará com que os feedbacks dos usuário finais cheguem logo e dessa forma as necessidades poderão ser refinadas no início do desenvolvimento. Essa prototipagem visa criar logo a interface de usuário pois assim é muito mais fácil fazer com que os usuários entendam como o sistema irá funcionar.

As etapas de abordagem são: Escrita dos requisitos preliminares, a projeção do protótipo, a utilização do protótipo pelo usuário e especifica as novas exigências, Especificação dos requisitos finais e finalmente o desenvolvimento do produto real.

A prototipagem Evolutiva é uma prototipagem muito diferente da Throwaway pois o seu principal objetivo é construir um protótipo muito robusto de forma que possa ser constantemente aperfeiçoado. A ideia por trás disso é que quando construído o protótipo vai se tornando cada vez mais parte do produto final. Ao desenvolver utilizando esse método o sistema sempre será refinado e reconstruído. Essa forma de criar protótipos têm uma vantagem sobre a de Throwaway na medida em que são sistemas se tornam funcionais. Embora possam não ter todas as funcionalidades que os usuários têm planejados, eles podem ser usados normalmente até que o sistema final seja entregue. Além do que os desenvolvedores podem focar em desenvolver as partes do sistema que mais lhe agradam ao invés de trabalhar no desenvolvimento de um sistema inteiro.

Desenvolvimento Rápido de Aplicações

O Desenvolvimento Rápido de Aplicações foi uma resposta aos processos desenvolvidos na década de 1970 e 1980, como a Análise de Sistemas Estruturada e método de projeto além de outros modelos de Cascata . Um problema com metodologias anteriores era que os aplicativos levavam tanto tempo para serem construídos que os requisitos tinham que ser mudado antes que o sistema estivesse terminado, resultando assim em sistemas inadequados ou até mesmo inutilizáveis. Outro problema foi a suposição de que a fase de análise metódica requisitos por si só, identificaria todos os requisitos críticos.

RAD envolve o desenvolvimento iterativo e a construção de protótipos . Em 1990, em seu livro *Desenvolvimento RAD aplicação rápida*, James Martin documentava a sua interpretação da metodologia, mais recentemente o termo e sua sigla têm vindo a ser usado em um sentido mais amplo, geralmente que engloba uma variedade de métodos que visam o desenvolvimento de aplicações em alta velocidade, tais como o uso de frameworks de software de variados tipos, como frameworks de aplicação web.

Começando com as ideias de Brian Gallagher, Balchin Alex, Barry Boehm e Shultz Scott, James Martin que desenvolveram a abordagem de desenvolvimento rápido de aplicações durante a década de 1980 na IBM e finalmente formalizada por uma publicação do livro *Rapid Application Development* em 1991.

As fases do RAD tem duas versões:

Primeira versão:

Planejamento de Requisitos fase: Combina elementos do sistema de planejamento e as fases de Análise de Sistemas do Ciclo de Vida do Desenvolvimento de Sistemas (SDLC). Usuários, gestores e membros da equipe de TI discutem e acordam sobre as necessidades do negócio, escopo do projeto, restrições e requisitos do sistema. Ela termina quando a equipe concorda com as principais questões e obtém autorização da gerência para continuar.

Fase do projeto de Usuário: Durante esta fase, os usuários interagem com os analistas de sistemas e desenvolvem modelos e protótipos que representam todos os processos do sistema, entradas e saídas. Os grupos ou subgrupos RAD geralmente usam uma combinação de conjuntos de desenvolvimento de aplicativos (JAD), técnicas e ferramentas CASE para traduzir as necessidades dos usuários em modelos de trabalho. Design de usuário é um processo contínuo interativo que permite aos usuários compreender, modificar e eventualmente aprovar um modelo de trabalho o sistema que atenda às suas necessidades.

Fase de construção: Centra-se sobre o programa e a tarefa de desenvolvimento de aplicativos parecida com o SDLC. Em RAD, no entanto, os usuários continuam participando e ainda podem sugerir mudanças ou melhorias como telas reais ou como os relatórios serão desenvolvidos. As suas tarefas são de programação e desenvolvimento de aplicação, codificação, unidade de integração e testes do sistema.

Transição de fase: Assemelha-se às tarefas finais na fase de implementação SDLC, incluindo a conversão de dados, testes, passagem para o novo sistema, e treinamento de usuários. Comparado com os métodos tradicionais, todo o processo é comprimida. Como resultado, o novo sistema é construído, entregue e colocado em operação muito mais cedo. As suas tarefas são a conversão de dados, testes em larga escala, mudança do sistema, treinamento de usuários.

Segunda versão:

Modelagem de Negócios: O fluxo das informações entre as funções de negócio é definido respondendo a perguntas como: Quais as informações que impulsionam o processo de negócio?, O que é a informação gerada?, O que a gera?, Onde faz o movimentação da informação?, Como a informação é processada? e assim por diante.

Modelagem de Dados: As informações coletadas a partir de modelagem de negócios são refinadas em um conjunto de objetos de dados (entidades) que serão necessárias para suportar o negócio. Os atributos (personagem de cada entidade) são identificados e a relação entre esses objetos de dados (entidades) será definida.

Modelagem de Processos: Os objetos de dados definidos na fase de modelagem de dados são transformados para conseguir o fluxo de informações necessárias para implementar uma função de negócio. Descrições de processamento são criadas para adicionar, modificar, apagar ou recuperar um objeto de dados.

Geração de aplicações: Ferramentas automatizadas são usadas para facilitar a construção do software, mesmo que eles usam as técnicas da linguagem de programação da quarta geração, que é uma linguagem de programação concebida com um propósito específico em mente, tais como o desenvolvimento de software de negócios comercial.

Teste e Entrega: Muitos dos componentes de programação já foram testados. Isto reduz o tempo de teste geral. Mas novos componentes devem ser testados como também todas as interfaces.

A mudança da sessão tradicional de desenvolvimento baseado entre cliente e servidor para abrir desenvolvimento sem sessão e colaborativa como a Web 2.0 tem aumentado a necessidade de repetições mais rápidas através das fases do processo de desenvolvimento de software. Isso, juntamente com o crescente uso de frameworks open source e produtos de desenvolvimento comercial do núcleo, para muitos desenvolvedores reacendeu o interesse em encontrar uma nova ferramenta na metodologia RAD.

Todos os tipos de RAD têm o potencial para fornecer um bom quadro para rápido desenvolvimento de produtos com melhor qualidade de software, mas a implementação bem sucedida e os benefícios muitas vezes dependem do tipo de projeto, cronograma, ciclo de lançamento de software e cultura corporativa. Porém também pode ser de interesse que alguns dos maiores fabricantes de software, como Microsoft e IBM não usam extensivamente o RAD no desenvolvimento de seus principais produtos e na sua maior parte, eles ainda se baseiam principalmente em tradicionais metodologias cascatas com algum grau de espiral.

Conclusão

Em (RAD) as técnicas estruturadas e de prototipagem são especialmente usadas para definir os usuários, os requisitos além de projetar o sistema final. O processo de desenvolvimento começa com o desenvolvimento de modelos preliminares de dados e de processos comerciais usando técnicas estruturais. Na etapa seguinte, os requisitos são verificados usando protótipos, eventualmente, para refinar os dados e modelos de processos. Estes estágios são repetidos iterativamente. O resultado disso são alguns requisitos de negócios que são combinados com a parte técnica para serem utilizados na construção de novos sistemas. Abordagens RAD pode implicar compromissos em termos de funcionalidade e desempenho em troca de permitir um desenvolvimento mais rápido e facilitar a manutenção da aplicação.

Referencias Bibliográficas

BlueInk:

<http://www.blueink.biz/RapidApplicationDevelopment.aspx>

Wikipedia:

http://en.wikipedia.org/wiki/Software_prototyping

http://en.wikipedia.org/wiki/Iterative_and_incremental_development

http://en.wikipedia.org/wiki/Rapid_application_development