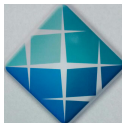


Desenvolvimento Rápido de Aplicações em Python

Aula 01

Evandro J.R. Silva¹

¹ Bacharelado em Ciência da Computação
Estácio Teresina



Sumário

1 Introdução

2 Fases

- James Martin
- James Kerr

3 Vantagens e Desvantagens

4 Devo usar RAD no meu projeto?

5 FIM

Introdução

- Desenvolvimento Rápido de Aplicações ou *Rapid Application Development* (RAD)
— o que é isso?

Introdução

- Desenvolvimento Rápido de Aplicações ou *Rapid Application Development* (RAD)
 - o que é isso?
 - É um **modelo de processo de desenvolvimento de software incremental**.

Introdução

- Desenvolvimento Rápido de Aplicações ou *Rapid Application Development* (RAD)
— o que é isso?
 - É um **modelo de processo de desenvolvimento de software incremental**.
 - Dentro da área de Engenharia de Software, RAD se encaixa nas metodologias ágeis.

Introdução

- Desenvolvimento Rápido de Aplicações ou *Rapid Application Development* (RAD)
 - o que é isso?
 - É um **modelo de processo de desenvolvimento de software incremental**.
 - Dentro da área de Engenharia de Software, RAD se encaixa nas metodologias ágeis.
 - Seu foco está na prototipação, para que haja rápido e constante *feedback* dos clientes/usuários → o que contribui para incrementar a aplicação.

Introdução

- Desenvolvimento Rápido de Aplicações ou *Rapid Application Development* (RAD)
— o que é isso?
 - É um **modelo de processo de desenvolvimento de software incremental**.
 - Dentro da área de Engenharia de Software, RAD se encaixa nas metodologias ágeis.
 - Seu foco está na prototipação, para que haja rápido e constante *feedback* dos clientes/usuários → o que contribui para incrementar a aplicação.
 - Diferente do clássico Modelo em Cascata, o RAD se propõe a ser bem mais flexível.

Introdução

- Desenvolvimento Rápido de Aplicações ou *Rapid Application Development* (RAD)
— o que é isso?
 - É um **modelo de processo de desenvolvimento de software incremental**.
 - Dentro da área de Engenharia de Software, RAD se encaixa nas metodologias ágeis.
 - Seu foco está na prototipação, para que haja rápido e constante *feedback* dos clientes/u-suários → o que contribui para incrementar a aplicação.
 - Diferente do clássico Modelo em Cascata, o RAD se propõe a ser bem mais flexível.
- A origem desse termo se deu em 1991, a partir de um livro publicado por James Martin.
 - Importante não confundir com o termo homônimo criado na década de 1970, porém este consistia em um **processo de desenvolvimento** de software.

Fases do RAD

- Existem duas principais abordagens para o RAD. Uma proposta por James Martin e a outra proposta por James Kerr.

Fases do RAD

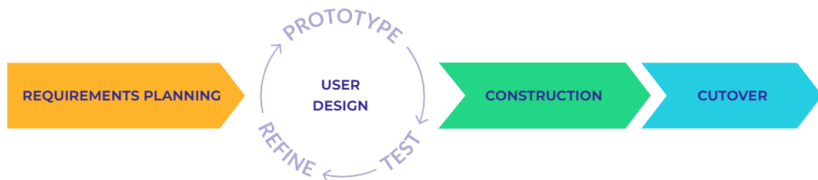
- Existem duas principais abordagens para o RAD. Uma proposta por James Martin e a outra proposta por James Kerr.
- Fases por James Martin:
 - 1 Planejamento de Requisitos;
 - 2 Design do Usuário;
 - 3 Construção;
 - 4 Transição.

Fases do RAD

- Existem duas principais abordagens para o RAD. Uma proposta por James Martin e a outra proposta por James Kerr.

- Fases por James Kerr:
 - 1 Modelagem de Negócios;
 - 2 Modelagem de Dados;
 - 3 Modelagem de Processos;
 - 4 Geração da Aplicação;
 - 5 Teste e Modificação.

Fases do RAD por James Martin



Fases do RAD por James Martin

1 Planejamento de Requisitos

- Nesta fase os usuários e a equipe de desenvolvimento discutem sobre qual o objetivo se quer alcançar. Estão inclusos nesta discussão o escopo do projeto, as restrições e requisitos de sistema.
- A fase termina quando se chega a um acordo sobre os pontos principais do sistema.
- É importante lembrar que a flexibilidade é um dos pilares do RAD. Portanto, o time deve estar sempre preparado para ajustar ou mudar os requisitos durante a prototipagem.

Fases do RAD por James Martin

1 Planejamento de Requisitos

2 Design do Usuário

- Nesta fase os desenvolvedores começam trabalhando com um protótipo. O objetivo é mostrar alguma coisa (funcionalidade) para o cliente o mais rápido e barato possível.
- É importante lembrar que o protótipo não vai ter incluso todos os requisitos e cenários possíveis.
- Assim que estiver pronto, o protótipo é apresentado ao(s) cliente(s)/usuário(s). Os *feedback* são coletados e então os requisitos podem ser ajustados.
- A equipe de desenvolvimento continua nessa etapa de prototipagem até que o(s) cliente(s)/usuário(s) fiquem satisfeitos.

Fases do RAD por James Martin

1 Construção

- Nesta fase o time de desenvolvimento já sabe o que precisa ser feito (após os vários *feedbacks* do(s) cliente(s)/usuário(s) na fase anterior).
- O sistema é desenvolvido com o foco em qualidade, escalabilidade e manutenibilidade.
- O(s) cliente(s)/usuário(s) ainda continuam participando, dando *feedback* à medida em que as *features* são implementadas.
- É possível ter pequenos ajustes (*fine tuning*) nesta fase.

Fases do RAD por James Martin

1 Construção

2 Transição

- Fase final, o que inclui testes (como o de aceitação), treinamento do usuário, *deploy*, etc.

Fases do RAD por James Kerr

1 Modelagem de Negócios

- "Nesta fase, serão definidos quais as características do negócio, isto é, define-se um fluxo de informações, que contém dados sobre: o que é gerado, quem gerou, para onde vão os dados e como é feito o processamento."

Fases do RAD por James Kerr

1 Modelagem de Negócios

2 Modelagem de Dados

- Nesta fase, o foco é sobre as funcionalidades.
- As informações extraídas na fase anterior são refinadas e, a partir disso, são extraídos os principais objetos de dados (e suas características) a serem processados pelo sistema.

Fases do RAD por James Kerr

1 Modelagem de Negócios

2 Modelagem de Dados

3 Modelagem de Processos

- "Os objetos de dados definidos na modelagem de dados são transformados para conseguir o fluxo necessário para implementar uma função do negócio. Descrições do processamento são criadas para adicionar, modificar, descartar ou recuperar um objeto de dados."

Fases do RAD por James Kerr

1 Modelagem de Negócios

2 Modelagem de Dados

3 Modelagem de Processos

4 Geração da Aplicação

- Fase em que o time de programação utilizará de ferramentas que ajudem a acelerar o processo de desenvolvimento. Isto inclui reusabilidade de componentes, ferramentas automatizadas, etc.

Fases do RAD por James Kerr

1 Modelagem de Negócios

2 Modelagem de Dados

3 Modelagem de Processos

4 Geração da Aplicação

5 Teste e Modificação

- Fase onde a aplicação passa pelos devidos testes.

Vantagens e Desvantagens

■ Vantagens

- Ciclo de desenvolvimento bastante curto (entre 60 e 90 dias);

Vantagens e Desvantagens

■ Vantagens

- É possível completar mais projetos dentro do tempo e orçamento estimados;

Vantagens e Desvantagens

■ Vantagens

- Andamento do projeto mais perceptível/visível, devido aos protótipos;

Vantagens e Desvantagens

■ Vantagens

- Flexibilidade durante o desenvolvimento;

Vantagens e Desvantagens

■ Vantagens

- Maior envolvimento do usuário, o que resulta em um software que seja mais relevante às suas necessidades, atendendo expectativas.

Vantagens e Desvantagens

■ Vantagens

- Custo e risco reduzidos. Mudanças inevitáveis, ou problemas percebidos *tardamente*, possuem um alto custo em um desenvolvimento no modelo Cascata.

Vantagens e Desvantagens

■ Desvantagens

- Seres humanos naturalmente rejeitam novidades. Um dos motivos é o risco de se tentar algo novo. Desta forma são exigidos profissionais experientes em RAD para que algum cliente se sinta mais confortável.

Vantagens e Desvantagens

■ Desvantagens

- Falta de ênfase nos **requisitos não funcionais**.

Vantagens e Desvantagens

■ Desvantagens

- Menor controle. Uma das vantagens do RAD é seu processo flexível e adaptável. Em um cenário ideal o desenvolvimento consegue se adaptar rapidamente a problemas e oportunidades. Entretanto, há uma troca (*trade-off*) inevitável entre flexibilidade e controle, de modo que mais de um significa menos do outro. Se um projeto valoriza mais o controle, RAD deixa de ser apropriado.

Vantagens e Desvantagens

■ Desvantagens

- Pode acontecer de a equipe focar demais nos protótipos, resultando em negligência quanto ao design do sistema. Desta forma, uma aplicação acaba sendo construída tendo um menor desempenho.

Vantagens e Desvantagens

■ Desvantagens

- Falta de escalabilidade. O modelo RAD tem maior foco sobre times de desenvolvimento pequenos a médios. Um projeto maior, que exige uma equipe maior, não vai ser tão bem gerenciada, devido ao controle diminuído e falhas de comunicação.

Vantagens e Desvantagens

■ Desvantagens

- Comprometimento dos usuários. Como o RAD prevê uma colaboração estrita entre o time de desenvolvimento e os usuários, uma quantidade grande de usuários (por exemplo, funcionários de uma empresa) acaba dificultando todo o processo. Além disso, etapas em que seja necessário o envolvimento de algum usuário especialista pode se tornar difícil por causa da agenda apertada do especialista.

Devo usar RAD no meu projeto?

- Sim, se
 - a aplicação possuir requisito muito bem definidos;
 - o sistema pode ser modularizado (reuso de código);
 - pode-se fazer o uso de APIs;
 - o desempenho não é o mais importante;
 - o time de desenvolvimento é relativamente pequeno.

Devo usar RAD no meu projeto?

■ Não, se

- o desempenho é essencial;
- o produto for de larga escala e/ou são muitos os envolvidos;
- o sistema não pode ser modularizado;
- não houver acesso regular aos usuários.

Fim

FIM!

Esta aula será postada no canal da disciplina!