

INTRODUÇÃO AO RAD

1.1 - INTRODUÇÃO AO RAD
1.2 - FASES DO RAD

Informe



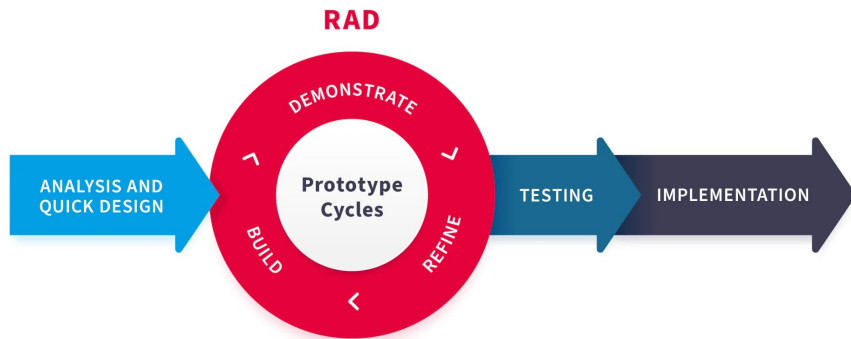
- O material do conteúdo digital da disciplina estará disponível na plataforma digital.

1. Contextualização

- O desenvolvimento rápido de aplicações RAD é uma metodologia de desenvolvimento de software com foco na entrega em um período muito inferior ao do ciclo de desenvolvimento tradicional de software.
- Não se trata de uma entrega final, mas, sim, de um **protótipo do software**. Para que isso seja possível, é feito um planejamento mínimo para obter um protótipo rápido.
- Na metodologia **RAD**, existe uma concentração no desenvolvimento dos principais módulos funcionais do sistema. Essa versão inicial é o protótipo.
- Trata-se de um modelo adaptativo, uma vez que o desenvolvimento é feito em iterações em que mudanças podem ser realizadas a partir dos comentários do usuário.
- A ênfase está na criação rápida de um protótipo, em vez de um planejamento detalhado.

1. Contextualização

- A RAD também pode ser aplicada para aperfeiçoar o treinamento prático de estudantes de computação, auxiliando-os em seus futuros empregos. Isso porque os estudantes podem aplicar o conhecimento adquirido nas aulas para desenvolver sistemas em etapas, conforme é proposto pela RAD. Como será mostrado mais adiante, o fator humano é um importante requisito para a aplicação dessa metodologia, então a sua aplicação para treinar recursos humanos pode acelerar a curva de aprendizado dentro de um curto período.



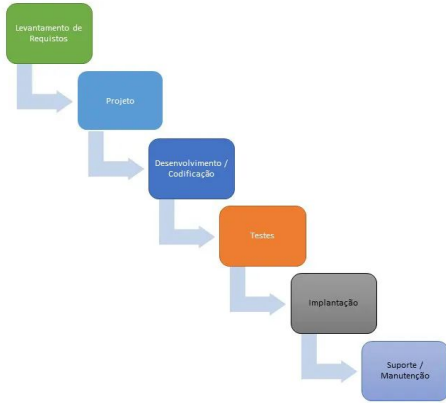
1. Contextualização



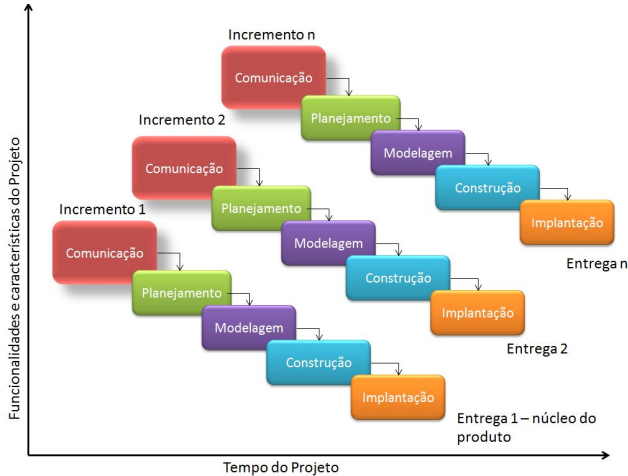
Um dos aspectos mais importantes deste modelo é **garantir** que os **protótipos** desenvolvidos sejam reutilizáveis para o projeto do sistema, ou seja, a ideia não é criar unidades descartáveis, mas isso não contradiz o fato de o protótipo ser flexível.

A RAD foca no desenvolvimento rápido por meio de **iterações frequentes e feedback contínuo**.

1. Contextualização



Os projetos RAD seguem o modelo iterativo e incremental. As equipes de desenvolvimento são **pequenas**, compostas por desenvolvedores, analistas de negócio e representantes de clientes.



Um dos aspectos mais importantes deste modelo é garantir que os protótipos desenvolvidos sejam reutilizáveis para o projeto do sistema, ou seja, a ideia não é criar unidades descartáveis. Mas isso não contradiz o fato de o protótipo ser flexível.

RAD

Características

1.1. Características do RAD

Envolvimento do usuário: ênfase na colaboração entre os usuários finais e os desenvolvedores, permitindo que os usuários forneçam feedback e validem os protótipos de software à medida que são desenvolvidos.

👉 **Ciclos de desenvolvimento curtos:** Concentra-se em ciclos de desenvolvimento curtos que permitem que os desenvolvedores entreguem um sistema funcional em um curto período de tempo, geralmente entre 60 e 90 dias.

👉 **Prototipagem rápida:** Ênfase na criação rápida de protótipos de software que podem ser rapidamente refinados e iterados. Os protótipos são usados para validar as necessidades dos usuários e as funcionalidades do software.

👉 **Equipe pequena e multifuncional:** Geralmente usa uma equipe pequena e multifuncional composta por desenvolvedores, designers, testadores e usuários finais, que trabalham juntos em estreita colaboração.

👉 **Reutilização de componentes:** a metodologia RAD enfatiza a reutilização de componentes de software existentes sempre que possível para acelerar o desenvolvimento.

👉 **Entrega incremental:** a metodologia RAD enfatiza a entrega incremental do software, permitindo que os usuários finais usem o software desde o início e forneçam feedback para informar o desenvolvimento futuro.

RAD

Características

Elementos

1.1.1. Elementos Fundamentais

- 1 **USO DE FERRAMENTAS PARA DAR SUPORTE AO DESENVOLVIMENTO:** O uso de ferramentas **CASE**(do inglês **Computer-Aided Software Engineering**) facilita a automação no desenvolvimento de sistemas. Isso é obtido através de recursos como geração de código e verificação automática de erros de consistência. As ferramentas CASE, portanto, são usadas para gerar protótipos, dando, assim, suporte ao desenvolvimento iterativo, possibilitando que os usuários finais acompanhem a evolução do sistema à medida que ele está sendo construído.
- 2 **METODOLOGIA BEM DEFINIDA:** É seguido um processo formal de desenvolvimento com atividades em etapas e entregas intermediárias. As tarefas são organizadas de modo a não negligenciar nenhum dos aspectos pré-acordados, e as técnicas são documentadas para garantir que uma tarefa seja executada da maneira correta.
- 3 **PESSOAS:** Deve haver treinamento das pessoas tanto na metodologia de trabalho como no uso das ferramentas. As tarefas devem ser distribuídas por pequenas equipes, que devem trabalhar bem juntas.
- 4 **GESTÃO:** O gerenciamento do projeto deve ser feito com rapidez. Isso é obtido através de oficinas de Planejamento de Requisitos e Design de Sistema para extrair rapidamente os requisitos dos usuários. Além disso, deve ser feita alocação de tempo fixo (**Timebox**) para entregar iterativamente o sistema para os usuários.

RAD

Características

Elementos

Princípios

1.1.1. Elementos Fundamentais

👉 **ENVOLVIMENTO ATIVO DOS USUÁRIOS:** A metodologia RAD reconhece que o envolvimento do usuário é necessário para reduzir problemas caros de obtenção de requisitos. Além disso, os usuários podem rejeitar completamente os sistemas, se não estiverem suficientemente envolvidos no desenvolvimento. No centro da abordagem da RAD, estão as oficinas de design de aplicativos conjuntos (JAD) e planejamento de requisitos conjuntos.

👉 **EQUIPES PEQUENAS COM PODER DE DECISÃO:** As vantagens da elaboração de equipes pequenas estão na redução de ruídos de comunicação e na minimização de atrasos devido à burocracia que a hierarquia de uma metodologia tradicional impõe. Em relação aos ruídos de comunicação, os canais que tratam dessa área aumentam proporcionalmente ao tamanho da equipe, portanto equipes pequenas evitam a distorção e o conflito na comunicação. A respeito da redução do tempo, empoderar a equipe aumenta as chances de cumprir os prazos por causa da responsabilidade de tomada de decisão. As equipes têm o poder de tomar decisões sobre o design (embora as mudanças sejam reversíveis).

👉 **ENTREGA FREQUENTE DE PRODUTOS:** Diferentemente das metodologias de desenvolvimento tradicionais, em que os projetos podem levar muito tempo para serem concluídos, a RAD procura reduzir o tempo de desenvolvimento. Portanto, prazos mais curtos para o desenvolvimento são uma característica importante. Em vez de se concentrar no processo, a RAD tem como premissa a entrega de produtos que satisfazem os requisitos funcionais.

1.1.1. Elementos Fundamentais

👉 **DESENVOLVIMENTO INCREMENTAL E ITERATIVO:** Outro princípio fundamental do RAD é que os sistemas evoluem de forma incremental em cada iteração. A cada nova iteração, surgem novos requisitos que são incorporados ao sistema. Desse modo, os sistemas evoluem através da prototipagem iterativa. Existe um entendimento no RAD que a especificação de requisitos é um processo não determinístico e que evolui à medida que desenvolvedores e usuários interagem com o protótipo do sistema.

👉 **ABORDAGEM TOP-DOWN:** Uma vez que, na metodologia RAD, os requisitos não precisam ser completamente definidos logo no início do projeto, eles são especificados em um nível apropriado ao conhecimento disponível no momento. Estes são então elaborados através de prototipagem incremental. Os sistemas são elaborados e confeccionados à medida que o conhecimento cresce. Além disso, como se trata de uma abordagem de “cima para baixo” caracterizada por um curto período, todas as decisões são consideradas reversíveis rapidamente.

👉 **UTILIZAÇÃO DE FERRAMENTAS DE AUTOMAÇÃO (CASE):** Trata-se de usar programas que facilitem a automação de processos, criação de diagramas, realização de testes e quaisquer tarefas que facilitem as entregas dentro dos prazos pré-estabelecidos e, obviamente, com qualidade. Além disso, essas ferramentas facilitam a reutilização de componentes que podem ser usados ao longo do projeto.

RAD

Características

Elementos

Princípios

Origem

2. Origem do RAD



O modelo RAD foi introduzido pelo consultor de tecnologia e autor James Martin que escreveu o livro **"Rapid Application Development"** em 1991. O livro de Martin descreveu uma abordagem para o desenvolvimento de software que enfatizava a entrega rápida e iterativa de um sistema funcional, bem como a colaboração entre os usuários finais e os desenvolvedores.

A RAD foi a precursora do gerenciamento **ágil de projetos**. As características de prototipagem rápida e ciclos de liberação e iterações mais curtos fortaleceram o posicionamento da RAD como um método eficaz no desenvolvimento de software, tornando-se cada vez mais popular entre as empresas ágeis que procuram métodos que acompanhem o crescimento de suas necessidades comerciais e de clientes. Trata-se de uma metodologia orientada pelo feedback do usuário, e não por um planejamento detalhado e caro.

RAD

Características

Elementos

Princípios

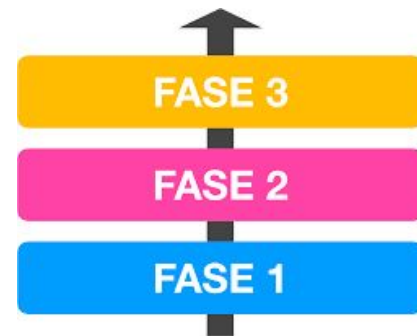
Origem

**Tipos de Projeto
RAD**

3. Tipos de Projetos RAD

FASEADO

Um projeto em fases é aquele distribuído por um longo período. Esses projetos são normalmente iniciados por um workshop JAD. As fases subsequentes do projeto são geralmente organizadas em termos de entrega e demonstração de protótipos incrementais. O objetivo é refinar continuamente o protótipo, tornando-o algo que seja entregue no final do timebox.



INTENSIVO

No tipo de projeto intensivo, uma equipe de desenvolvedores e usuários trabalham por um curto período (algumas semanas) e, ao final desse tempo, espera-se que produza um produto que seja utilizável.



RAD

Características

Elementos

Princípios

Origem

**Tipos de Projeto
RAD**

**Como iniciar um
projeto RAD**

3.1. Como iniciar um projeto RAD?

- Uma das formas de iniciar o projeto RAD é através da aplicação da metodologia Joint Application Development (JAD).
- Trata-se de uma metodologia na qual usuários e analistas projetam o sistema juntos, sob uma liderança em oficinas de trabalho.
- A ideia é potencializar o resultado do desenvolvimento através de dinâmicas de grupo.
- Ou seja, definir os objetivos e as aplicações do sistema, desde a geração de telas até a geração de relatórios.
- Tem como princípios: dinâmica de grupo; recursos audiovisuais; processo organizado e racional; a escolha do local; documentação com a abordagem WYSIWIG – “O que você vê é o que você obtém”.

RAD

Características

Elementos

Princípios

Origem






**Tipos de Projeto
RAD**

**Como iniciar um
projeto RAD**

**Ferramentas e
Técnicas**

4. Ferramentas

A RAD precisa ser suportada por ferramentas que auxiliem no desenvolvimento das aplicações rapidamente. Entre as categorias de ferramentas que dão suporte à RAD para desenvolver projetos de software estão:

-  Integração de dados
-  Ambientes de desenvolvimento
-  Ferramentas de coleta de requisitos
-  Ferramentas de modelagem de dados
-  Ferramentas de geração de código



4. Técnicas

TÉCNICA	PARTICULARIDADE
Modelo CBD	O método que descreve como componentes antigos podem ser reutilizados com os novos.
RepoGuard	É um framework para integração de ferramentas de desenvolvimento com repositórios de código-fonte.
Adição dinâmica ágil	Técnicas usadas para integração do ágil para tornar o projeto mais adaptável.
Método baseado em camadas para desenvolvimento rápido de software	Baseado em camadas que segue o XP.
Análise de projeto de sistema baseado em simulação	Desenvolvimento de ferramentas ágeis baseadas em simulação.
Desenvolvimento de aplicativos multiusuário em ambiente distribuído rapidamente.	Middleware de comunicação.
Programação extrema	Adição de reutilização ao XP.

RAD

Características

Elementos

Princípios

Origem


**Tipos de Projeto
RAD**

**Como iniciar um
projeto RAD**


**Ferramentas e
Técnicas**

Aplicação


5. Aplicação


 **Desenvolvimento de protótipos:** Os ciclos de desenvolvimento curtos e a prototipagem rápida permitem que você crie um protótipo funcional em um curto período de tempo.


- É muito útil para a compreensão do sistema
- Serve de demonstração para os clientes
- É mais flexível para mudanças
- Quando está mais evoluído, pode ser integrado ao produto completo para uma entrega mais rápida da versão final

 **Necessidades dos usuários em constante mudança:** se você está desenvolvendo um software para atender às necessidades dos usuários em constante mudança, a entrega incremental permite que você entregue o software aos usuários desde o início, permitindo que eles forneçam feedback constante para orientar o desenvolvimento.

5. Aplicação

 **Prazos apertados:** se você precisa entregar um software funcional em um curto período de tempo, a metodologia RAD pode ser uma boa opção. Os ciclos de desenvolvimento curtos e a entrega incremental permitem que você entregue um software funcional em um curto período de tempo, enquanto ainda pode continuar a melhorá-lo com o tempo.

 **Equipe pequena:** se você tem uma equipe pequena de desenvolvedores e precisa entregar um software funcional em um curto período de tempo. A equipe pequena e multifuncional pode trabalhar em estreita colaboração para entregar o software rapidamente e iterativamente.

 **Estudo Prático:** A RAD também pode ser aplicada para aperfeiçoar o treinamento prático de estudantes de computação, auxiliando-os em seus futuros empregos. Isso porque os estudantes podem aplicar o conhecimento adquirido nas aulas para desenvolver sistemas em etapas, conforme é proposto pela RAD. Como será mostrado mais adiante, o fator humano é um importante requisito para a aplicação dessa metodologia, então a sua aplicação para treinar recursos humanos pode acelerar a curva de aprendizado dentro de um curto período.

- BERGER, H.; BEYNON-DAVIES, P.: **The utility of rapid application development in large-scale, complex projects**. Information Systems Journal, 2009. 19 (6), 549– 570.
- bottlepy. Consultado em meio eletrônico em: 4 ago. 2020.
- cherrypy. Consultado em meio eletrônico em: 4 ago. 2020.
- Django. Consultado em meio eletrônico em: 4 ago. 2020.
- flask. Consultado em meio eletrônico em: 4 ago. 2020.
- Fitzgerald, B. **A Preliminary Investigation of Rapid Application Development in Practice, Proceedings of 6th International Conference on Information Systems Methodologies**, editors Wood-Harper AT, Jayarantna N., Wood J R G, pp. 77–87, 1998.
- Kerr, J.; Hunter, R. **Inside RAD: How to Build Fully Functional Computer Systems in 90 Days or Less**. New York: McGraw-Hill, 1994.
- Kivy. Consultado em meio eletrônico em: 4 ago. 2020.
- Martin, J. **Rapid Application Development**, Macmillan, USA, 1991.
- matplotlib. Consultado em meio eletrônico em: 4 ago. 2020.
- NAZ, R.; KHAN, M. N. A. **Rapid applications development techniques**: A critical review, Int. J. Softw. Eng. its Appl., vol. 9, nº. 11, pp. 163–176, 2015.
- NumPy. Consultado em meio eletrônico em: 4 ago. 2020.
- Pandas. Consultado em meio eletrônico em: 4 ago. 2020.
- PyQt. Consultado em meio eletrônico em: 4 ago. 2020.
- PySide. Consultado em meio eletrônico em: 4 ago. 2020.
- Python 3.8.5 documentation. Consultado em meio eletrônico em: 4 ago. 2020.
- Spyder. Consultado em meio eletrônico em: 4 ago. 2020.
- Tcl Developer Xchange. Consultado em meio eletrônico em: 4 ago. 2020.
- The Falcon Web Framework. Consultado em meio eletrônico em: 4 ago. 2020.
- turbogears. Consultado em meio eletrônico em: 4 ago. 2020.
- wxPython. Consultado em meio eletrônico em: 4 ago. 2020.