



Rational Unified Process **(RUP)**

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br

Revisão



Software

Instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados;



Engenharia de software

Engenharia de software é a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software.





Processo de software

São as atividades envolvidas na produção de um sistema de software. Modelos de processos de software são representações abstratas desses processos.



Modelos de processo

Modelos gerais de processo descrevem a organização dos processos de software. Exemplos desses modelos gerais incluem o modelo em cascata, o desenvolvimento incremental e o desenvolvimento orientado a reuso.



Modelos de processo

Uma metodologia de processo genérica para engenharia de software compreende cinco atividades:

- Comunicação
- Planejamento
- Modelagem
- Construção
- Emprego



Modelos de processo

Também podemos usar os conceitos de :

Engenharia de requisitos é o processo de desenvolvimento de uma especificação de software. As especificações destinam-se a comunicar as necessidades de sistema dos clientes para os desenvolvedores do sistema.

Processos de projeto e implementação estão relacionados com a transformação das especificações dos requisitos em um sistema de software executável. Métodos sistemáticos de projeto podem ser usados como parte dessa transformação.



Modelos de processo

Também podemos usar os conceitos de :

Validação de software é o processo de verificação de que o sistema está de acordo com sua especificação e satisfaz às necessidades reais dos usuários do sistema.

Evolução de software ocorre quando se alteram os atuais sistemas de software para atender aos novos requisitos. As mudanças são contínuas, e o software deve evoluir para continuar útil



Processos

Processos devem incluir atividades para lidar com as mudanças. Podem envolver uma fase de prototipação, que ajuda a evitar más decisões sobre os requisitos e projeto. Processos podem ser estruturados para o desenvolvimento e a entrega iterativos, de forma que mudanças possam ser feitas sem afetar o sistema como um todo.

Rational Unified Process - RUP Processo Unificado da Rational



Rational Unified Process - RUP

O *Rational Unified Process* — RUP (KRUTCHEN, 2003) é um exemplo de modelo de processo moderno, derivado de trabalhos sobre a UML e o *Unified Software Development Process* associado (RUMBAUGH, et al., 1999; ARLOW e NEUSTADT, 2005).

Um acrônimo para a expressão “*Unified Modeling Language*” ou simplesmente “Linguagem de Modelagem Unificada”, o UML é uma linguagem de notação para uso em projetos de sistema.



Rational Unified Process - RUP

RUP é um bom exemplo de processo híbrido. Ele reúne elementos de todos os modelos de processo genéricos (modelo cascata, incremental), ilustra boas práticas na especificação e no projeto (comunicação, planejamento, modelagem, construção, emprego) e apoia a prototipação e a entrega incremental.



Definição

É um processo proprietário de Engenharia de software criado pela Rational Software Corporation, adquirida pela IBM, então RUP ganhou o nome de IRUP IBM Rational Unified Software (prém o nome mais conhecido é RUP), ele utiliza uma abordagem de orientação a objetos em sua concepção e é projetado e documentado utilizando a notação UML para ilustrar os processos em ação.

Um acrônimo para a expressão “*Unified Modeling Language*” ou simplesmente “Linguagem de Modelagem Unificada”, o UML é uma linguagem de notação para uso em projetos de sistema.



Rational Unified Process - RUP

O RUP reconhece que os modelos de processo convencionais apresentam uma visão única do processo. Em contrapartida, o RUP é normalmente descrito em três perspectivas:

1. Uma perspectiva dinâmica, que mostra as fases do modelo ao longo do tempo.
2. Uma perspectiva estática, que mostra as atividades realizadas no processo.
3. Uma perspectiva prática, que sugere boas práticas a serem usadas durante o processo.



Rational Unified Process - RUP

A maioria das descrições do RUP tenta combinar as perspectivas estática e dinâmica em um único diagrama (KRUTCHEN, 2003). Por achar que essa tentativa torna o processo mais difícil de ser compreendido, uso descrições separadas de cada perspectiva.



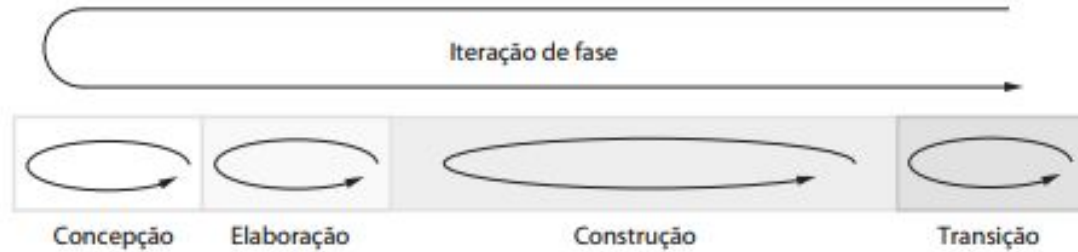
Rational Unified Process - RUP

O RUP utiliza pequenos ciclos de projeto que correspondem à uma iteração e que resultam em um incremento no software. Iterações referem-se a passos e incrementos à evolução do produto.

Cada ciclo possui 4 fases: Concepção, Elaboração, Construção e Transição

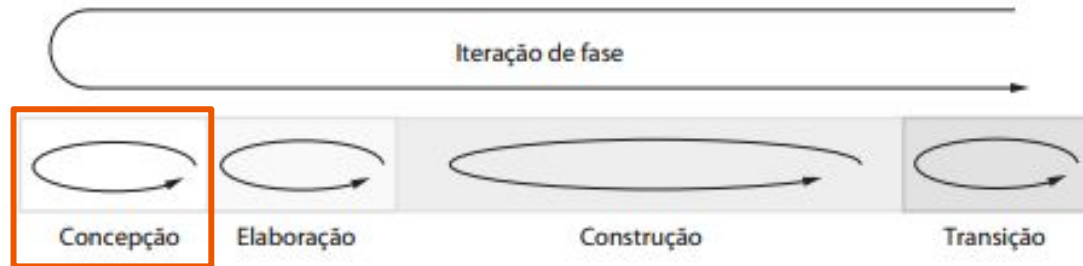
Fases no *Rational Unified Process*

Fases no Rational Unified Process



Fases no *Rational Unified Process*

Fases no Rational Unified Process



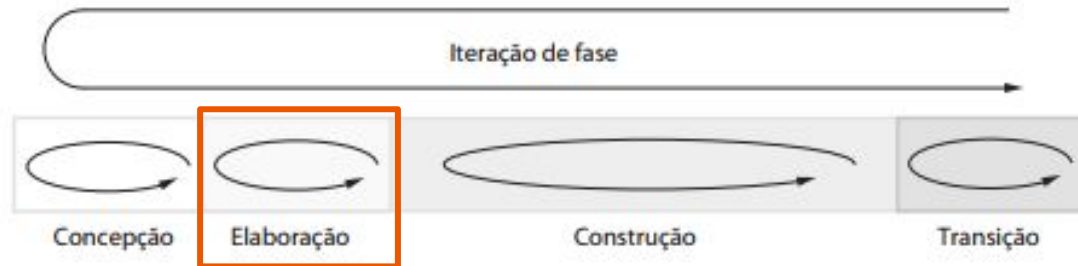


Concepção

O objetivo da fase de concepção é estabelecer um *business case* para o sistema. Você deve identificar todas as entidades externas (pessoas e sistemas) que vão interagir com o sistema e definir as interações. Então, você deve usar essas informações para avaliar a contribuição do sistema para o negócio. Se essa contribuição for pequena, então o projeto poderá ser cancelado depois dessa fase.

Fases no *Rational Unified Process*

Fases no Rational Unified Process





Elaboração

As metas da fase de elaboração são desenvolver uma compreensão do problema dominante, estabelecer um *framework* da arquitetura para o sistema, desenvolver o plano do projeto e identificar os maiores riscos do projeto. No fim dessa fase, você deve ter um modelo de requisitos para o sistema, que pode ser um conjunto de casos de uso da UML, uma descrição da arquitetura ou um plano de desenvolvimento do software.

Fases no *Rational Unified Process*

Fases no Rational Unified Process





Construção

A fase de construção envolve projeto, programação e testes do sistema. Durante essa fase, as partes do sistema são desenvolvidas em paralelo e integradas. Na conclusão dessa fase, você deve ter um sistema de software já funcionando, bem como a documentação associada pronta para ser entregue aos usuários

Fases no *Rational Unified Process*

Fases no Rational Unified Process



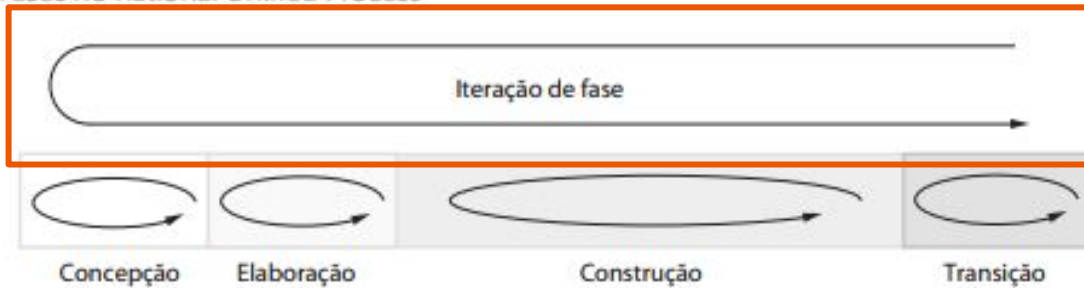


Transição

A fase final do RUP implica transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários e em seu funcionamento em um ambiente real. Isso é ignorado na maioria dos modelos de processo de software, mas é, de fato, uma atividade cara e, às vezes, problemática. Na conclusão dessa fase, você deve ter um sistema de software documentado e funcionando corretamente em seu ambiente operacional

Fases no *Rational Unified Process*

Fases no Rational Unified Process





Iteração

No RUP, a iteração é apoiada de duas maneiras. Cada fase pode ser executada de forma iterativa com os resultados desenvolvidos de forma incremental. Além disso, todo o conjunto de fases também pode ser executado de forma incremental.



RUP - Visão estática

A visão estática do RUP prioriza as atividades que ocorrem durante o processo de desenvolvimento. Na descrição do RUP, essas são chamadas *workflows*. Existem seis *workflows* centrais, identificadas no processo, e três *workflows* de apoio.



RUP - Visão estática

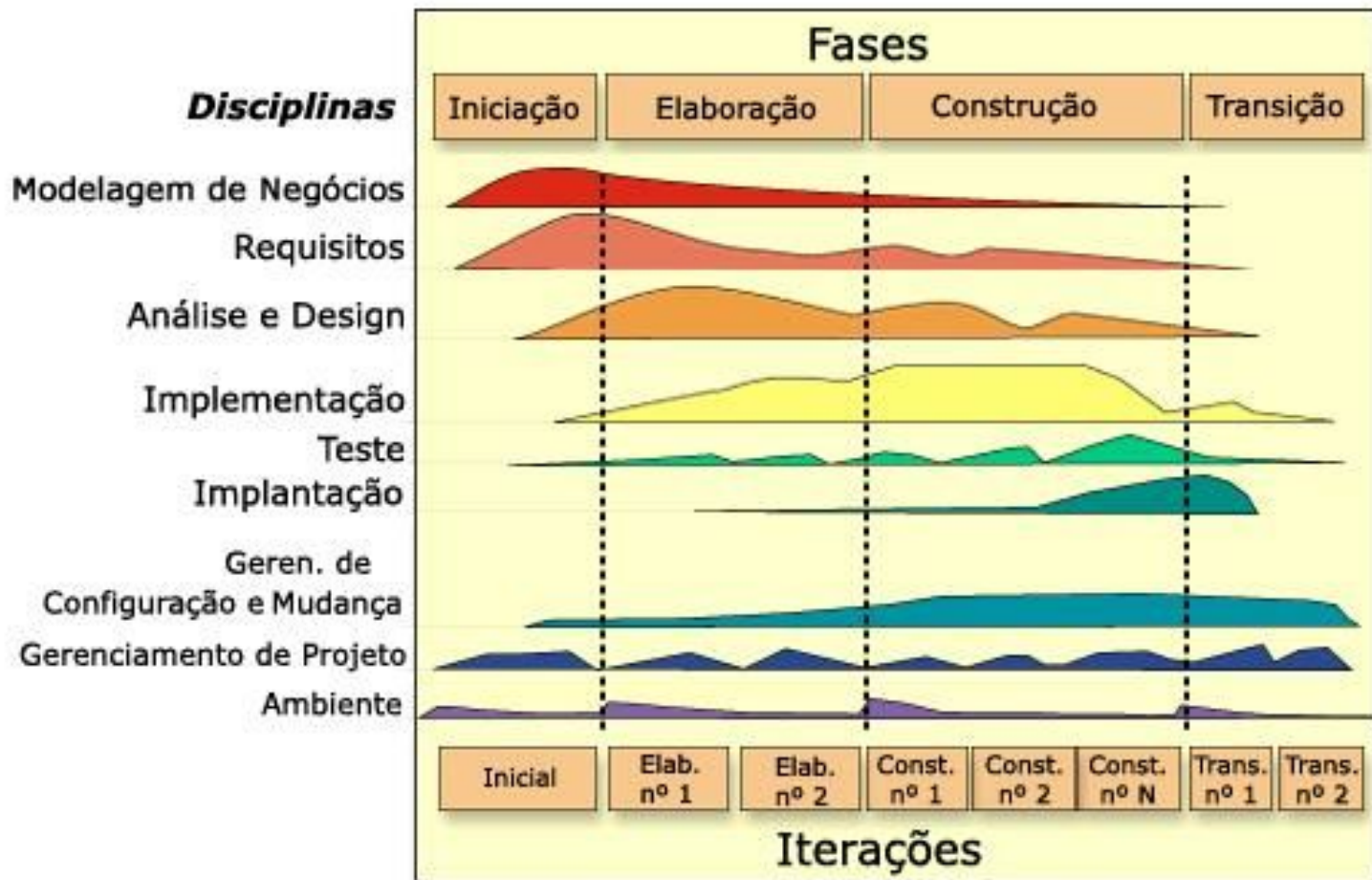
O RUP foi projetado em conjunto com a UML, assim, a descrição do *workflow* é orientada em torno de modelos associados à UML, como modelos de sequência, modelos de objetos etc.

WORKFLOW	DESCRIÇÃO
Modelagem de negócios	Os processos de negócio são modelados por meio de casos de uso de negócios.
Requisitos	Atores que interagem com o sistema são identificados e casos de uso são desenvolvidos para modelar os requisitos do sistema.
Análise e projeto	Um modelo de projeto é criado e documentado com modelos de arquitetura, modelos de componentes, modelos de objetos e modelos de sequência.
Implementação	Os componentes do sistema são implementados e estruturados em subsistemas de implementação. A geração automática de código a partir de modelos de projeto ajuda a acelerar esse processo.
Teste	O teste é um processo iterativo que é feito em conjunto com a implementação. O teste do sistema segue a conclusão da implementação.
Implantação	Um <i>release</i> do produto é criado, distribuído aos usuários e instalado em seu local de trabalho.
Gerenciamento de configuração e mudanças	Esse <i>workflow</i> de apoio gerencia as mudanças do sistema (veja o Capítulo 25).
Gerenciamento de projeto	Esse <i>workflow</i> de apoio gerencia o desenvolvimento do sistema (veja os capítulos 22 e 23).
Meio ambiente	Esse <i>workflow</i> está relacionado com a disponibilização de ferramentas apropriadas para a equipe de desenvolvimento de software.



Visão Estática x Dinâmica

A vantagem de proporcionar visões estáticas e dinâmicas é que as fases do processo de desenvolvimento não estão associadas a *workflows* específicos. Ao menos em princípio, todos os *workflows* do RUP podem estar ativos em todas as fases do processo. Nas fases iniciais, provavelmente, maiores esforços serão empenhados em workflows, como modelagem de negócios e requisitos, e, nas fases posteriores, no teste e na implantação.





Perspectiva prática RUP

RUP descreve as boas práticas da engenharia de software que são recomendadas para uso no desenvolvimento de sistemas. Seis boas práticas fundamentais são recomendadas:

1. Desenvolver software iterativamente;
2. Gerenciar os requisitos;
3. Usar arquiteturas baseadas em componentes;
4. Modelar o software visualmente;
5. Verificar a qualidade do software;
6. Controlar as mudanças do software.



Desenvolver software iterativamente

Planejar os incrementos do sistema com base nas prioridades do cliente e desenvolver os recursos de alta prioridade no início do processo de desenvolvimento.



Gerenciar os requisitos

Documentar explicitamente os requisitos do cliente e acompanhar suas mudanças. Analisar o impacto das mudanças no sistema antes de aceitá-las.



Usar arquiteturas baseadas em componentes

Estruturar a arquitetura do sistema em componentes.



Modelar o software visualmente.

Usar modelos gráficos da UML para apresentar visões estáticas e dinâmicas do software.



Verificar a qualidade do software

Assegurar que o software atenda aos padrões de qualidade organizacional.



Controlar as mudanças do software

Gerenciar as mudanças do software, usando um sistema de gerenciamento de mudanças e procedimentos e ferramentas de gerenciamento de configuração.



Conclusão

O RUP não é um processo adequado para todos os tipos de desenvolvimento, como, por exemplo, desenvolvimento de software embutido.

As inovações mais importantes do RUP são a separação de fases e workflows e o reconhecimento de que a implantação de software em um ambiente do usuário é parte do processo.



Conclusão

As fases são dinâmicas e têm metas. Os workflows são estáticos e são atividades técnicas que não são associadas a uma única fase, mas podem ser utilizadas durante todo o desenvolvimento para alcançar as metas específicas.



Exercícios

Resolva os exercícios propostos na lista de exercícios 2.



Bibliografia

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. 2. ed. Rio de Janeiro: Campus, 2006.

PRESSMAN, Roger S. Engenharia de software. 5. Ed. Rio de Janeiro: McGraw-Hill, 2002.

SOMMERVILLE, I. Engenharia de software. 10. Ed. São Paulo, SP: Addison Wesley, 2018

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. Rio de Janeiro: Campus, 2006.