

Universidade do Contestado – UNC
Unidade Universitária de Mafra
Otávio Rodolfo Piske
Curso de Sistemas de Informação – 5ª Fase

RUP – Rational Unified Process

MAFRA
2003
Otávio Rodolfo Piske

1 - Introdução

O RUP (Rational Unified Process) é uma metodologia para desenvolvimento de software criada pela Rational Software, IBM, SofTeam, Unisys, Nihon Unisys, Alcatel e Q-Labs. O RUP pode ser encontrado na forma de um software, fornecido pela Rational Software, e como um conjunto de processos. Neste trabalho iremos cobrir apenas aspectos relativos ao conjunto de processos referentes ao RUP, incluindo:

- * conceitos
- * best practices (melhores práticas)
- * fases de desenvolvimento

2 - RUP - Conceitos

Como citado anteriormente, o RUP é mais do que um softwares para auxiliar no desenvolvimento é uma metodologia de desenvolvimento, com uma estrutura formal e bem definida. Como qualquer metodologia, é composta de conceitos, práticas e regras.

Um dos principais pilares do RUP é o conceito de *best practices* (melhores práticas), que são regras/práticas que visam reduzir o risco (existente em qualquer projeto de software) e tornar o desenvolvimento mais eficiente. O RUP define seis *best practices*, sendo elas:

- * desenvolver iterativamente
- * gerenciar requerimentos
- * utilizar arquiteturas baseadas em componentes
- * modelar visualmente
- * verificação contínua de qualidade
- * controle de mudanças

O RUP, ainda, entrelaça o conceito de *best practices* em quatro definições, sendo elas:

- * funções: grupos de atividades executadas.
- * disciplinas: áreas de esforço na engenharia de software.
- * atividades: definições de como (objetos/artefatos) é construído e avaliado.
- * objetos/artefatos: resultado do trabalho, produzido ou modificado durante o processo.

Além destas definições, esta metodologia de desenvolvimento divide o processo de desenvolvimento de software em quatro fases (as quais serão discutidas com mais detalhes posteriormente). São elas:

- * concepção: definição do escopo do projeto.
- * elaboração: elaboração básica do software.
- * construção: desenvolvimento.
- * transição:

3 - RUP – *Best Practices*

O RUP tenta diminuir os riscos do desenvolvimento e efetivamente deixar o desenvolvimento mais eficiente, através de seis práticas básicas (conhecidas por *best practices*) a serem executadas durante todo o processo de desenvolvimento.

3.1 - *Desenvolver Iterativamente*

Desenvolver iterativamente significa desenvolver em ciclos. Cada ciclo contém um objetivo que deve ser alcançado (lançamento de um pre-release ou beta, correção de um bug, etc).

Esta prática acaba dando ao RUP uma série de vantagens, como a possibilidade de identificar/modificar requerimentos com mais facilidade; integração progressiva (quase continua) de elementos ao software, ocasionando uma melhora no descobrimento e endereçamento de riscos; desenvolvimento iterativo provê aos gerentes maneiras de fazer mudanças táticas aos produtos; etc.

3.2 - Gerenciar Requerimentos

Gerenciamento de requerimentos provê uma maneira prática de produzir, organizar, comunicar e organizar os requerimentos de um projeto. Adicionalmente, os casos de uso e cenários descritos nos processo são uma excelente forma de capturar e assegurar requisitos. O gerenciamento de recursos acarreta um melhor controle sobre projetos complexos, além de maior qualidade e redução de custos.

O RUP é uma metodologia dirigida-a-casos-de-uso (use-driven-case), de modo que é possível utilizar os mesmos casos de uso definidos no sistema como base para o resto do processo.

3.3 - Utilizar Arquiteturas Baseadas em Componentes

Foca o desenvolvimento na modularização, através do uso de componentes, de modo a criar um sistema flexível, adaptável, intuitivamente entendível e reutilizável. O RUP entende componentes como módulos não triviais e/ou subsistemas com uma função clara e específica. Entre os benefícios podemos citar a facilidade para identificar, isolar, manipular e desenvolver componentes é maior do que para um sistema inteiro; componentes podem ser desenvolvidos com a reutilização em mente; etc.

3.4 - Modelar Visualmente

A modelagem visual permite melhor entender não só a concepção e a complexidade do sistema, mas também “dimensionar” (no sentido de qual a forma do sistema), além de facilitar a identificação e solução de problemas.

3.5 - Verificação Contínua de Qualidade

O RUP não toma a qualidade como responsabilidade de apenas

uma pessoa ou grupo, mas como uma responsabilidade de todos os integrantes do projeto.

A qualidade é focada especialmente em duas áreas:

- Qualidade de produto: a qualidade do produto sendo desenvolvido (software or sistema) e todas as partes envolvidas (componentes, subsistemas, arquitetura, etc).
- Qualidade de processo: a qualidade dos processos dentro do projeto de desenvolvimento.

3.6 - Controle de Mudanças

Como resultado de um processo de desenvolvimento iterativo, muitas são as mudanças ocorridas no decorrer do projeto. Controlar as mudanças durante todo o projeto é prática fundamental para manter a qualidade do projeto.

4 - Fases de Desenvolvimento

O processo de desenvolvimento é dividido em ciclos, sendo que o ciclo de desenvolvimento é subdividido em 4 fases consecutivas.

Estas fases são concluídas tão logo uma *milestone* é alcançada. Uma *milestone* define uma etapa, na fase, na qual decisões críticas são feitas ou objetivos são alcançados.

4.1 - Concepção

Concepção inicial do sistema, aonde é feita uma discussão sobre o problema, definição do escopo do projeto, estimativa de recursos necessários para a execução do projeto, etc. É nesta fase que é apresentada o plano de projeto, caso de uso inicial e o glossário do projeto, entre outros.

4.2 - Elaboração

O propósito desta fase é analisar o domínio do problema, desenvolver o plano de projeto, estabelecer a fundação arquitetural e eliminar os elementos de alto risco.

Os elementos de risco a serem analisados, nesta fase, são os riscos de requerimentos, tecnológicos (referentes a capacidade das ferramentas disponíveis), de habilidades (dos integrantes do projeto) e políticos.

Esta é a fase mais crítica de todas, pois ao final desta fase a engenharia é considerada completa e os custos para modificação do sistema aumentam a medida que o projeto avança. Do ponto de vista administrativo, é ao final desta fase que um projeto deixa de ser uma operação de baixo risco e baixo custo para se tornar uma operação de alto risco e alto custo.

4.3 - Construção

Esta fase compreende a fase de modelagem e a fase de desenvolvimento em si, aquela em que o sistema é efetivamente programado. A fase de modelagem deve utilizar alguma notação definida pela UML.

4.4 - Transição

A partir desta fase, o sistema já está pronto, começa a implantação do sistema para o usuário (ou a comunidade de usuários do mesmo). Nesta fase é deve ser utilizado o lançamento de versões beta, operação paralela com o sistema legado, treinamento dos usuários e mantenedores do sistema, etc.

5 - Conclusão

O RUP prova ser um processo de desenvolvimento robusto e bem definido, embora bastante complexo/trabalhoso para projetos de software de pequeno porte, ele pode ser bem aproveitado para projetos aonde é preciso manter registro constante do fluxo do projeto.

6 - Bibliografia

Quadros, Moacir - Gerencia de Projetos de Softwares: Técnicas e Ferramentas, Visual Books.

Rational Software - Best Practices for Software Development Teams, www.rational.com.

Rational Software - Rational Unified Process, www.rational.com.

The Rational Edge - RUP and XP, Part I - Finding Common Ground, <http://therationaledge.com>.