

Rational Unified Process: Reduzindo riscos nos projetos de software

Laís Marques Santos¹

Resumo

Este artigo traz um apanhado geral sobre uma metodologia que aumenta a produtividade nos projetos de software e busca minimizar os riscos que geralmente ocorrem nesses projetos através do desenvolvendo dos sistemas de forma incremental. A metodologia no presente artigo permite ao cliente visualizar e acompanhar melhor o processo de criação do projeto. Será mostrado como isso é possível através de uma boa estruturação dos processos dentro do projeto e de uma divisão das atividades que permitem controlar melhor as alterações, indicando o impacto delas e reduzindo os riscos que acontecem dentro do projeto durante seu desenvolvimento. Ao final, será feito um comparativo entre as duas maiores metodologias incrementais do mercado.

Palavras-chave: Metodologia; Processos; Projeto; Rup, XP.

ABSTRACT

This article provides an overview of a methodology that enhances productivity in software projects and seeks to minimize the risks that often happen in these projects by developing systems incrementally. The methodology in this article allows the client to better visualize and track the process of project creation. It will show how this is possible through good structural processes within the project and a division of activities that allow better control changes, indicating their impact and reducing the risks that happen within the project during its development. Finally, a comparison is made between the two major methods of incremental market.

Keywords: Methodology; Processes; Projects; Rup, XP.

¹ SANTOS, Laís Marques. Pós-graduanda em Sistemas de Informação para Web pela Faculdade de Administração e Negócios de Sergipe; nível superior de curta duração em Desenvolvimento de Aplicações Web pela Universidade Tiradentes; graduada em Bacharelado em Matemática pela Universidade Federal de Sergipe. Atualmente trabalha como Desenvolvedora Web na Infox Tecnologia da Informação Ltda. Email: lais@infox.com.br

INTRODUÇÃO

A atividade de desenvolvimento de software possui um alto grau de risco. Essa atividade já gerou e continua gerando grandes prejuízos, pois muitos projetos de software são iniciados e não conseguem ser terminados, e outros são terminados, mas consomem prazos e orçamentos bem acima do que foi estimado no início do projeto.

O êxito de um projeto de desenvolvimento de software começa com seu devido planejamento e com a escolha de uma metodologia compatível com suas características. Esse planejamento deve estruturar o processo de desenvolvimento em torno dos recursos disponíveis visando à entrega de um produto de qualidade que atenda às expectativas e necessidades do cliente dentro do prazo estabelecido.

O presente artigo dará uma visão geral de uma metodologia criada por uma grande empresa do mercado: a Rational. Essa metodologia chamada de Processo de Software Unificado (RUP), busca a redução de riscos desenvolvendo sistemas de forma incremental e utilizando-se de boas práticas. Ela divide o desenvolvimento de um projeto em ciclos, visando redução de riscos (e custos) nos casos de alterações de requisitos ou funcionalidades, diferentemente das metodologias tradicionais que são divididas em etapas seguidas estritamente de forma seqüencial de modo que uma alteração no requisito pode levar o projeto a sofrer enormes alterações.

O artigo está estruturado em outras sete seções. A seção 1 apresenta um apanhado geral sobre PU que foi usada de base para a criação do RUP. Nas seções 2, 3, 4 e 5 serão apresentados, respectivamente, o que é o RUP, suas fases, fluxos e artefatos. Na seção seguinte será feito um comparativo do RUP com outra grande metodologia incremental existente no mercado, o XP, no intuito de fazer levantamentos sobre vantagens e desvantagens de cada processo. Já a última seção apresenta as considerações finais do trabalho.

2. PROCESSO UNIFICADO

Assim que se iniciou o desenvolvimento de softwares, por falta de uma metodologia específica para essa nova área, convencionou-se utilizar as metodologias já existentes no mercado. O problema é que essas metodologias não

eram adequadas para a construção de softwares e muitos projetos terminaram por 'naufragar'. Utilizava-se muito o modelo em cascata, onde cada etapa do ciclo de vida é realizada integralmente e de uma só vez assim como, por exemplo, a construção de um prédio, mas isso não era adequado para softwares.

O Processo Unificado (PU) foi criado para ser um processo de desenvolvimento de software visando à construção de sistemas orientados a objetos que aborda a construção de projeto de forma mais realística já que é mais apropriada para desenvolvimento de softwares.

O PU é um processo iterativo e adaptativo de desenvolvimento que pode ser personalizado de acordo com as necessidades específicas e recursos disponíveis para cada projeto. Devido à maneira organizada e consistente que permite conduzir um projeto, esse processo vem ganhando cada vez mais adeptos.

O ciclo de vida iterativo do PU é baseado em refinamentos e incrementos sucessivos com a finalidade de convergir para a produção de um sistema adequado. Em cada interação o produto é um pouco mais incrementado, baseando-se na experiência obtida nas iterações anteriores e no feedback do usuário. Cada interação pode ser considerada um mini-projeto de duração fixa, sendo que cada um destes inclui suas próprias atividades de análise de requisitos, projeto, implementação e testes. As atividades são repetidas quantas vezes forem preciso, em ciclos organizados de modo que possibilita detectar se o projeto está no caminho certo ou, por exemplo, se alguma mudança na arquitetura central deve ser feita.

2. O QUE É O RUP?

O RUP, abreviação de Rational Unified Process (Processo de Software Unificado), é uma metodologia criada pela Rational Software Corporation para viabilizar que grandes projetos de software sejam bem sucedidos. Essa metodologia que é uma instância específica e detalhada do Processo Unificado, recentemente virou um produto comercializado pela IBM que adquiriu a Rational em fevereiro de 2003.

Este processo capacitado para a Web unifica toda a equipe de desenvolvimento de software e otimiza a produtividade de cada membro da equipe proporcionando melhores práticas de software a todos os membros.

Um processo (de engenharia) de software é a definição de um conjunto completo de actividades necessárias para transformar os requisitos do utilizador num conjunto consistente de artefactos que representam um produto de software e, mais tarde, transformar alterações nesses requisitos num novo conjunto consistente de artefactos. (JACOBSON, 2000, p.34)

O processo do RUP é configurável e adaptável para qualquer projeto de software ou organização que desenvolva ou implante software e são apoiadas por ferramentas através da família Rational Suite.

O Rational Unified Process dispõe de suporte para técnicas orientadas a objeto. Os modelos do RUP são baseados nos conceitos de objeto e classes e nos relacionamentos existentes entre eles.

A característica fundamental do RUP é ser baseado em componentes, ou seja, o software desenvolvido é formado por componentes de software que se comunicam através de interfaces bem definidas. O padrão adotado para representação dos modelos é a *Unified Modeling Language* (UML). Lembrando que a UML é apenas uma linguagem para representação e não constitui um processo de desenvolvimento em si.

Através de uma abordagem disciplinada, o RUP busca atribuir e gerenciar tarefas e responsabilidades com o objetivo de proporcionar o desenvolvimento, dentro de custos e cronograma previsíveis, de softwares de alta qualidade que endereçam corretamente as necessidades de seus usuários finais.

O RUP baseia-se nas 6 melhores práticas de desenvolvimento de software moderno:

- Desenvolver interativamente e controladamente ao longo de todo o Ciclo de Desenvolvimento;
- Gerenciar requisitos;
- Modelar visualmente com UML;
- Utilizar arquiteturas componentizadas;
- Controlar continuamente a qualidade através de testes automatizados;

- Fazer controle de versões, configurações e mudanças.

3. FASES DO RUP

O RUP divide o projeto em quatro fases: concepção, elaboração, construção e transição. A passagem pelas quatro fases é chamada um ciclo de desenvolvimento e resulta na geração de um software.

3.1. Fase de Concepção

Na primeira fase do ciclo de vida do processo é possível delimitar o escopo do projeto através da compreensão do problema e da tecnologia. Após as análises realizadas nessa fase verifica-se a viabilidade ou não do projeto do ponto de vista do negócio, pois já é possível ter a noção de quanto o sistema irá custar e quanto ele trará de retorno.

3.2. Fase de Elaboração

Na fase de elaboração, é estabelecido a fundação de uma arquitetura sólida, o desenvolvimento do plano do projeto e a eliminação dos elementos de maior alto risco do projeto.

Ao final dessa segunda fase, é possível estimar custos, elaborar o cronograma e o plano de construção do sistema.

3.3. Fase de Construção

Na fase de construção, começa o desenvolvimento físico do software que é feito de forma iterativa e incremental.

Nessa terceira fase do ciclo de desenvolvimento, além do código, propriamente dito, também são produzidos os casos de teste e a documentação.

3.4. Fase de Transição

Na fase de transição, o software é finalmente disponibilizado aos usuários finais. Mas, após o sistema ser colocado nas mãos de seus usuários, sempre surgem questões que requerem algum desenvolvimento adicional, com a finalidade

de ajustar o sistema, corrigir alguns problemas identificados ou concluir algumas características propostas anteriormente, por isso esta fase é tipicamente iniciada com uma versão beta do sistema, que depois é substituída pelo sistema de produção.

4. FLUXOS DO RUP

Cada fase do RUP pode ainda ser dividida em iterações. Cada iteração passa pelos vários fluxos de trabalho do processo, embora com uma ênfase diferente em cada um deles, dependendo da fase.

São nove os fluxos de trabalho com os quais o RUP é composto, são eles:

1. *Modelagem de Negócio*: Descreve a estrutura e a dinâmica da empresa;
2. *Requisito*: Descreve o método baseado em casos de uso para identificar requisitos;
3. *Análise e Projeto*: Descreve as várias visões da arquitetura;
4. *Implementação*: Leva em consideração o desenvolvimento do software, o teste da unidade de erros;
5. *Teste*: Descreve casos de teste, procedimentos e medidas para acompanhamento de erros;
6. *Entrega*: Abrange a configuração do sistema a ser entregue;
7. *Gerenciamento da Configuração*: Controla as modificações e mantém a integridade dos artefatos do projeto;
8. *Gerenciamento de Projeto*: Descreve várias estratégias para o trabalho com um processo iterativo;
9. *Ambiente*: Abrange a infra-estrutura necessária para o desenvolvimento do sistema.

A figura 1, que também é conhecida como gráfico de baleais, mostra o fluxo de vida baseado no RUP, suas 4 (quatro) fases (Fases do Processo Unificado da Rational – PUR), distribuídas de acordo com o Modelo de Iterações proposto por este processo.

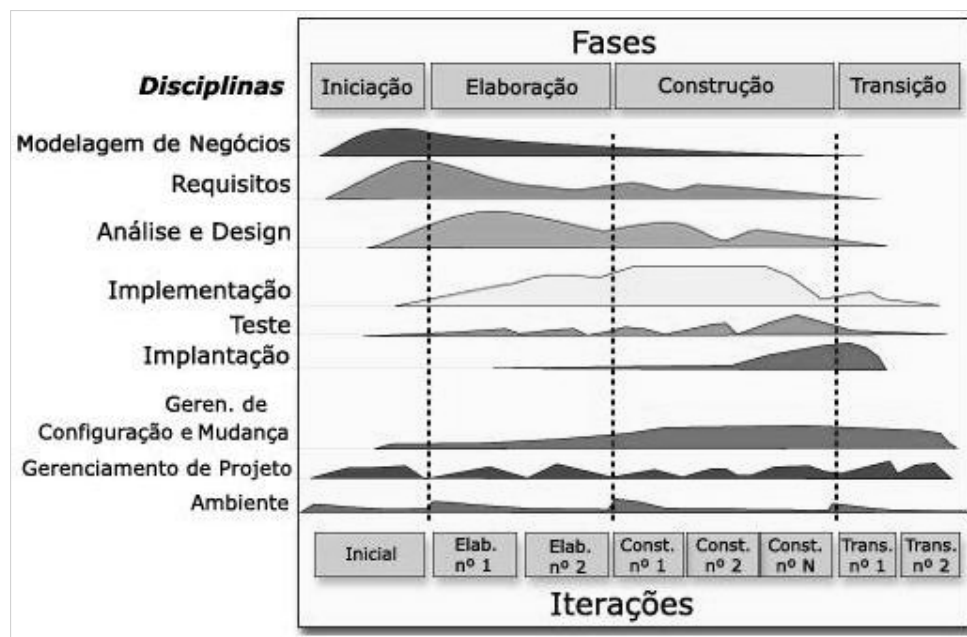


FIGURA 1 - Fases do Processo Unificado da Rational
 Fonte : <http://wthreex.com/rup/> . Acessado em: 8 de set. de 2010

Cada fluxo está associado a um conjunto de atividades e artefatos correlacionados.

5. ARTEFATOS

Um artefato é algum documento, relatório ou executável, que é produzido, manipulado ou consumido. Alguns artefatos são utilizados com a finalidade de direcionar a entrada para atividades subseqüentes, mantidos como recursos de referência sobre o projeto ou gerados em um formato como entregas contratuais.

O tipo mais importante de artefato do RUP são os modelos, que são uma simplificação da realidade criada para proporcionar uma melhor compreensão do sistema que está sendo criado. São nove os modelos que compõem o RUP: o modelo de negócio, o modelo de domínio, o modelo de caso de uso, o modelo de análise (opcional), o modelo de projeto, o modelo de processo (opcional), o modelo de implantação, o modelo de implementação e o modelo de teste.

Existem outros artefatos do RUP que são categorizados como artefatos de gerenciamento ou artefatos técnicos. Estes artefatos são divididos em quatro

conjuntos principais, são eles: conjunto de requisitos (descreve o que o sistema deve fazer), conjunto de projeto (descreve como o sistema é construído), conjunto de implementação (descreve como montar o sistema) e conjunto de implantação (fornece dados para a configuração de implantação).

A figura 2 ilustra os principais artefatos do RUP.

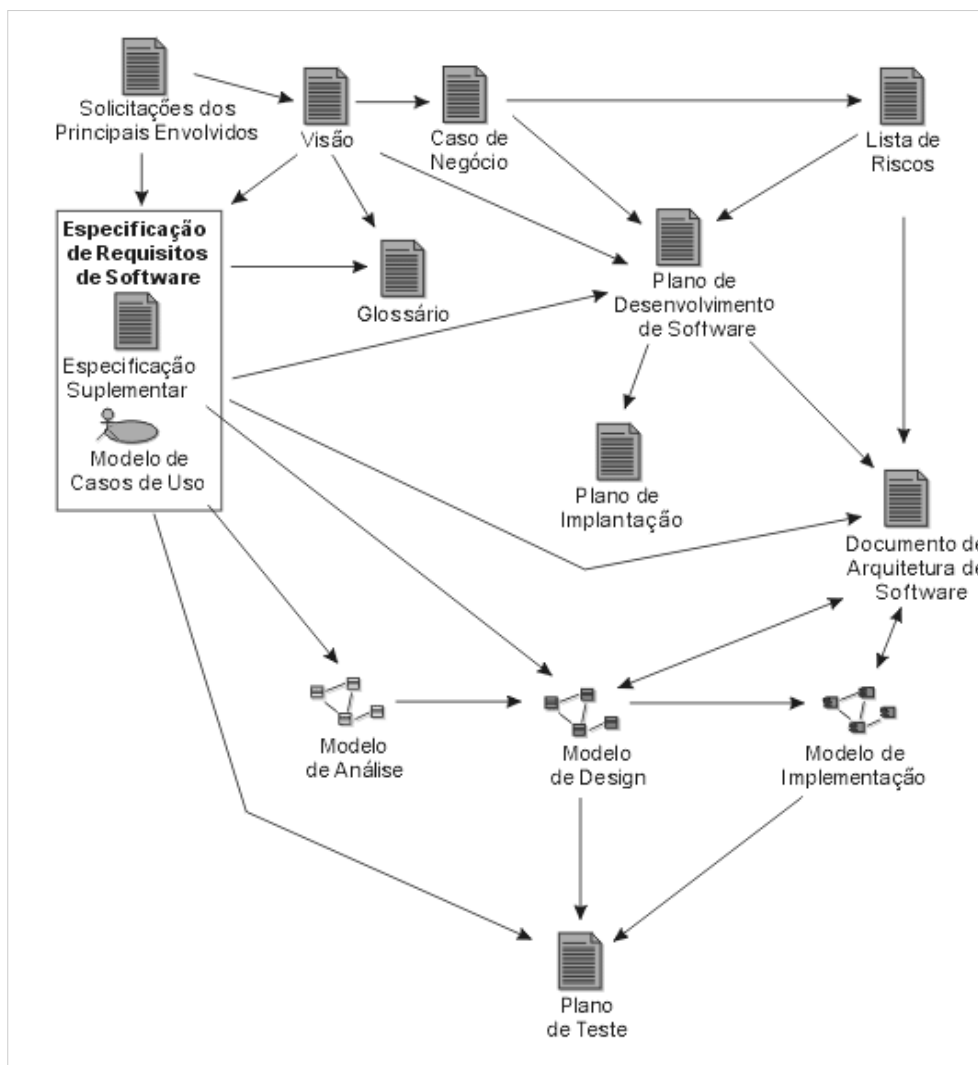


FIGURA 2 - Principais Artefatos do RUP

Fonte : <http://wthreex.com/rup/> . Acessado em: 8 de set. de 2010

6. COMPARAÇÃO ENTRE RUP E XP

Outra metodologia que busca a redução de riscos desenvolvendo sistemas de forma incremental é o Extreme Programming (XP).

O XP compartilha da mesma crença do Processo Unificado de que o desenvolvimento iterativo é a melhor forma de entregar aos clientes um software valioso. Ele baseia-se no rápido feedback em curtas iterações com liberações freqüentes. No XP o desenvolvimento de um projeto é dividido em ciclos, visando redução de riscos e custos nos casos de alterações de requisitos ou funcionalidades assim como o RUP.

Apesar de compartilhar muitas idéias de construção incremental de sistemas provenientes das metodologias iterativas, o XP enquadra-se mais ao conceito de metodologia ágil que possui dentre suas particularidades o fato de possuir iterações muito mais curtas.

6.1. Semelhanças

Independente de seus processos específicos, as duas metodologias, buscam reduzir riscos e aumentar a qualidade do produto gerado através dos seguintes princípios que possuem como semelhança:

- Baseada em Iterações;
- Versões pequenas (ciclos rápidos que reduzem riscos quando ocorrem alterações de requisitos ou funcionalidades);
- Facilitar a comunicação entre os vários interessados do projeto;
- Orientada a Cliente (envolvimento do cliente);
- Integração contínua;
- Definição de papéis
- Requer disciplina;
- Objetivam a simplicidade;
- Flexibilidade;
- Testes constantes.

Numa análise superficial, pode ser dito que essas duas metodologias tratam a dinâmica de desenvolvimento de software da mesma forma, mas apesar de

suas semelhanças, existem diferenças entre essas duas metodologias que serão apresentadas nessa seção.

6.2. Diferenças

Apesar das duas metodologias apresentarem os mesmos valores, elas buscam esses objetivos de forma diferente.

As duas são baseadas em iterações, mas possuem períodos diferenciados para cada iteração. Os períodos de iterações, da metodologia ágil (XP), geralmente são menores, medidos em semanas, enquanto que as iterações de processos iterativos (RUP) são geralmente medidos em meses.

O XP é concebido para equipes pequenas e médias, onde o projeto é tratado como um todo, não em subsistemas e, utiliza-se de programação em pares o que exige padrões de codificação e projetos simples a fim de evitar falha de comunicação e retrabalho. Já o RUP é concebido para grandes projetos onde as equipes são geralmente divididas para se desenvolver paralelamente, cada uma com um subsistema.

A comunicação dentro de um processo RUP é baseada em artefatos e documentações bem estruturadas. Já o XP é baseado em comunicação oral e direta, fugindo da documentação formal, o que restringe o uso de XP em projetos com grande distribuição geográfica. Um exemplo da disparidade, no ponto documentações, tem-se que no RUP faz-se uso da UML enquanto que o XP faz uso de uma modelagem informal.

O XP não especifica nenhuma ferramenta em específico para o processo enquanto que o RUP utiliza softwares, como o *Rational Rose*, que devem ser adquiridos da própria IBM, juntamente com a documentação.

Ao compararmos as definições das atividades (disciplinas) e os papéis, temos que o RUP faz uma divisão de tarefas de forma específica, possuindo 30 papéis agrupados em nove disciplinas (como visto na seção 4), enquanto que a divisão de papéis proposta pelo XP tem um caráter de “uso-geral”, sem atribuições específicas dentro das atividades. No XP é definido sete papéis distribuído dentro de quatro atividades básicas: produção de código, testes, *listening* (escutar o cliente), e desenho. Por não distinguir funções específicas dentro das atividades, no XP, o

programador não está necessariamente preso a determinada atividade, podendo se encaixar em outra, caso seja necessário.

O conceito de fases, como normalmente é descrito no Processo Unificado, é um pouco diferente no XP. As decisões descritas nas fases do Processo Unificado que definem os marcos ocorrem, porém não são chamadas especificamente de definidoras de fase.

No quesito responsabilidade de código, no RUP vê-se o código tendo determinados membros sendo responsáveis por ele, diferentemente do XP onde o código é tratado como coletivo onde qualquer programador que encontre algum problema, ou algum trecho que possa ser otimizado, tem permissão para fazê-lo. Por isso é necessário nos projetos que se utilizam do XP possuírem padrões de codificação, citados no primeiro parágrafo.

Apesar das duas metodologias indicam que o projeto completo não pode ser planejado em detalhe, o RUP indica modificação contínua dos planos, enquanto o XP propõe planejar em detalhes somente o futuro próximo.

A tabela 1 apresenta algumas semelhanças e diferenças entre as metodologias RUP e XP:

TABELA 1 - Semelhanças e diferenças entre o RUP e XP

Fonte: Diversas, 2010

	RUP	XP
Metodologia	Rigorosa	Ágil
Indicada para projetos	Grandes	Pequenos
Ciclos	Versões pequenas	Versões pequenas
Integração	Contínua	Contínua
Períodos de iterações	Semanas	Meses
Flexibilidade	Sim	Sim
Equipe	Grande. Dividida para desenvolver paralelamente	Pequenas e médias. Programação em pares
Disciplina	Requerida	Requerida
Qualidade de código	Não	Sim, seguindo padrões
Modelagem	UML	Informal
Comunicação	Artefatos e documentações	Oral e direta
Cliente	Participação ativa	Participação ativa
Atividades (disciplinas) /	Específica.	Sem atribuições

Papéis	30 papéis / 9 disciplinas	específicas. 7 papéis / 4 atividades
Testes	Constantes	Constantes
Utilização de softwares	Rational Rose	Não específica
Gerenciamento de risco	Sim	Sim

7. CONSIDERAÇÕES FINAIS

O RUP proporciona uma linguagem padrão para modelar projetos de sistemas, abrangendo desde aspectos conceituais até itens concretos. Sua correta utilização efetivamente possibilita maior produtividade, aumenta a qualidade do produto final e reduz custos.

Processos não podem ser comprados prontos e adotados a risca nas empresas, eles devem ser adequados à cultura organizacional. Por ser flexível e totalmente adaptável, o RUP pode ser adotado por organizações e em projetos de qualquer porte ou características, atendendo a necessidade de se configurar o processo para projetos distintos, de acordo com suas características. Mas, apesar de ser adaptável, seria mais recomendado o uso da metodologia ágil XP para projetos menores, como vimos na seção de comparativo.

Por ser dividido em fases bem definidas o RUP reduz a complexidade e permite visualizar e acompanhar melhor o processo de criação do projeto, controlando as alterações e indicando o impacto delas no projeto. Problemas e inconsistências são identificados e corrigidos a cada fase do fluxo e não somente no final do processo, pois cada iteração agrega novas integrações, e testes cumulativos, minimizando assim os riscos do projeto. Outro ponto importante que foi visto e que ajuda a reduzir os riscos do projeto foi que ao final de cada iteração código é distribuído, implantado e é validado pelo cliente.

Graças a essas divisões e controles, projetos que adotam a metodologia do RUP conseguem ser entregues dentro do prazo e nos custos esperados, além de melhorar a qualidade do software que produzem e aumentando, assim, a satisfação dos clientes.

REFERÊNCIAS BIBLIOGRÁFICAS

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML, guia do usuário**. Trad. Fábio Freitas da Silva. Rio de Janeiro: Campus, 2000.

CONALLEN, Jim. **Desenvolvimento de aplicações Web com UML**. Trad. Altair Dias Caldas de Moraes, Cláudio Belleza Dias. 2.ed. Rio de Janeiro: Campus, 2003.

FOWLER, Martin e SCOTT, Kendall. **UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos**. Trad. Vera Pezerico e Chistian Thomas Price. 2.ed. Porto Alegre: Bookman, 2000.

Guia de Introdução – Rational Unified Process. Disponível em: <<http://wthree.com/rup/>>. Acessado em: 08 de set. 2010.

KRUCHTEN, Phillippe. **INTRODUÇÃO AO RUP: Rational Unified Process**. 2.ed. Rio de Janeiro: Ciência Moderna, 2003.

MARTINS, José Carlos Cordeiro. Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML, In: _____ **Rational Unified Process**. 4.ed. Rio de Janeiro: Brasport, 2007. p. 192–248.

SCOTT, Kendall. **O Processo Unificado Explicado**. Porto Alegre: Bookman, 2003.

Rational Unified Process. Disponível em: <<http://www-01.ibm.com/software/br/rational/rup.shtml>>. Acessado em: 08 de set. 2010.