A Rational é bem conhecida pelo seu investimento em orientação a objetos. A empresa foi a criadora da Unified Modeling Language (UML), assim como de várias ferramentas que a suportam, sendo a mais conhecida o Rational Rose. O Rational Unified Process (RUP) é uma metodologia completa criada pela Rational para viabilizar que grandes projetos de software sejam bem sucedidos. O RUP é na verdade um produto composto de material de referência na forma de páginas HTML, descrevendo toda a metodologia.

**Princípios Básicos**

Um grande problema nos projetos atuais é o grande dinamismo e complexidade dos negócios nos dias de hoje. Cada vez mais os sistemas são complexos e precisam estar prontos em menos tempo. Mais do que isso, as necessidades mudam ao longo do tempo e a especificação de um sistema provavelmente será alterada durante seu desenvolvimento. Além disso, temos tecnologias novas (software e hardware) surgindo a cada dia. Algumas funcionam bem. Outras não. Visando atacar estes problemas, o RUP adota as seguintes premissas básicas:

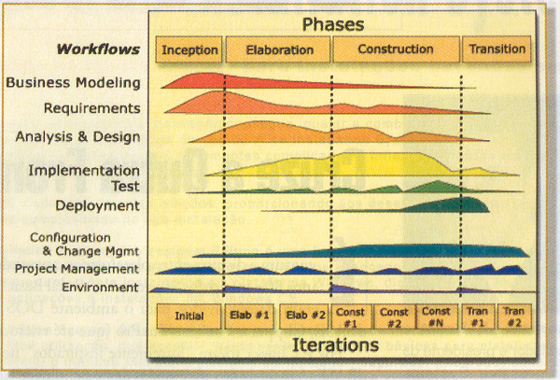
* Uso de iterações para evitar o impacto de mudanças no projeto,
* Gerenciamento de mudanças e
* Abordagens dos pontos de maior risco o mais cedo possível.

**Estrutura Básica do RUP**

A figura 1 apresenta os elementos básicos do RUP. Nesta metodologia, o projeto passa por 4 fases básicas. Estas fases são:

* Inception - entendimento da necessidade e visão do projeto,
* Elaboration - especificação e abordagem dos pontos de maior risco,
* Construction - desenvolvimento principal do sistema,
* Transition - ajustes, implantação e transferência de propriedade do sistema

Figura 1: Modelo Básico do RUP



Apesar de parecer um modelo em cascata, na verdade cada fase é composta de uma ou mais iterações, o que se assemelha a um modelo em espiral. Estas iterações são em geral curtas (1-2 semanas) e abordam algumas poucas funções do sistema. Isto reduz o impacto de mudanças, pois quanto menor o tempo, menor a probabilidade de haver uma mudança neste período para as funções em questão.

Além das fases e iterações, existem os workflows. Cada workflow é na verdade uma sequência de tarefas encadeadas e relacionadas a um aspecto importante do projeto, tal como análise do negócio, testes, etc. Os gráficos da figura mostram a ênfase de cada workflow em cada etapa do projeto.

**Conteúdo do RUP**

O que é o RUP afinal? Sabemos neste ponto que é uma metodologia completa e vimos sua estrutura básica. Porém o que ele oferece?

O RUP, além de uma metodologia, é um produto comercializado pela Rational, que é uma grande documentação baseada em hipertexto (HTML). Do conteúdo deste material, destaco os seguintes assuntos:

Workflows: Cada workflow é descrito em detalhe, apresentando passo a passo as tarefas, subprodutos a serem gerados e papéis de profissionais envolvidos. Cada tarefa, subproduto e papel é descrito em detalhe. Este modelo pode ser seguido à risca ou adaptado conforme sua necessidade específica.

Tarefas: Cada tarefa é descrita em detalhe, incluindo que papel é responsável por ela, a qual workflow ela pertence e quais são os subprodutos de entrada e saída.

Modelo de equipe: Os diversos papéis necessários no projeto são descritos em detalhe. Assim como no MSF, cada papel pode ser representado por mais de uma pessoa, ou uma pessoa pode ter mais de um papel, dependendo da carga de trabalho necessário. Porém o RUP aborda os papéis em um maior nível de detalhe. Ao todo são mais de 30. Exemplos de papéis são: analista de sistemas, analista de negócio, etc.

Modelos de documentos: O RUP apresenta modelos e exemplos para os diversos documentos (artifacts) gerados ao longo do projeto. Estes documentos são descritos em detalhe, assim como as tarefas que os geram e as que os utilizam. Para os usuários das ferramentas Rational, existe um recurso adicional e-coach, que orienta o usuário a usar ferramentas como o Requisite Pro passo a passo. Os documentos são totalmente compatíveis com a UML, o que reforça a questão de padronização.

**Como Adotar**

Com base nestes recursos a adoção do RUP pode ser feita de mais de uma maneira. Um extremo seria usar o RUP à risca, ou seja, aplicar todos os métodos e processos exatamente como são propostos. A vantagem desta abordagem é que nada deve ser alterado, pois o RUP é bem completo e detalhado. Porém existe um preço a ser pago, pois o RUP na íntegra é complexo. Esta abordagem implicaria em treinamentos, projetos piloto, etc. Propostas de projetos de adoção do RUP são descritos no próprio produto.

O extremo oposto seria adotar outro modelo de processo mais simples ou conhecido (o atual, se existir) e utilizar o material do RUP como fonte de referência complementar para assuntos não abordados em outro modelo como, por exemplo, os modelos de documentos.

A primeira abordagem é interessante para empresas que precisam de uma grande formalização do processo de desenvolvimento de software e cujo método atual seja totalmente inadequado ou inexistente. A segunda abordagem seria interessante para quem já tem alguma metodologia que considera adequada, mas que tem deficiência em alguma área como, por exemplo, suporte a UML. Soluções intermediárias também são possíveis.

Vimos de forma suscinta o que o RUP é e o que ele oferece. Para aqueles que desejam mais informações sobre o RUP, sugiro ler o livro “The Rational Unified Proccess, An Introduction” de Philippe Kruchten, disponível na Amazon. Para quem deseja ver o produto, uma cópia de avaliação pode ser solicitada a Rational (<http://www.rational.com>).

O **RUP**, abreviação de **Rational Unified Process** (ou Processo Unificado Racional), é um [processo](http://pt.wikipedia.org/wiki/Processo) proprietário de [Engenharia de software](http://pt.wikipedia.org/wiki/Engenharia_de_software) criado pela [Rational Software Corporation](http://pt.wikipedia.org/wiki/Rational_Software_Corporation), adquirida pela [IBM](http://pt.wikipedia.org/wiki/IBM), ganhando um novo nome **IRUP** que agora é uma abreviação de **IBM Rational Unified Process** e tornando-se uma [*brand*](http://pt.wikipedia.org/wiki/Brand) na área de Software, fornecendo técnicas a serem seguidas pelos membros da equipe de desenvolvimento de software com o objetivo de aumentar a sua produtividade no processo de desenvolvimento.

O RUP usa a abordagem da [orientação a objetos](http://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_orientada_a_objetos) em sua concepção e é projetado e documentado utilizando a notação [UML](http://pt.wikipedia.org/wiki/UML) (*Unified Modeling Language*) para ilustrar os processos em ação. Utiliza técnicas e práticas aprovadas comercialmente.

É um processo considerado pesado e preferencialmente aplicável a grandes equipes de desenvolvimento e a grandes [projetos](http://pt.wikipedia.org/wiki/Projeto), porém o fato de ser amplamente customizável torna possível que seja adaptado para projetos de qualquer escala. Para a [gerência do projeto](http://pt.wikipedia.org/wiki/Ger%C3%AAncia_de_projetos), o RUP provê uma solução disciplinada de como assinalar tarefas e responsabilidades dentro de uma organização de desenvolvimento de software.

O RUP é, por si só, um produto de software. É modular e automatizado, e toda a sua metodologia é apoiada por diversas ferramentas de desenvolvimento integradas e vendidas pela IBM através de seus "Rational Suites".

Métodos concorrentes no campo da engenharia de software incluem o "[Cleanroom](http://pt.wikipedia.org/wiki/Cleanroom" \o "Cleanroom)" (considerado pesado) e os [Métodos Ágeis](http://pt.wikipedia.org/wiki/Desenvolvimento_%C3%A1gil_de_software) (leves) como a [Programação Extrema](http://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_Extrema) (XP-Extreme Programming), [Scrum](http://pt.wikipedia.org/wiki/Scrum), [FDD](http://pt.wikipedia.org/wiki/FDD) e outros.

Segundo (Fowler, 2005 p39) e uma estrutura de processos fornecidos pela (Rational Unified Process) fornece um vocabulário e uma vaga estrutura sobre processos define o que e como fazer ele e basicamente uns processos interativos.

Segundo (Sommerville, 2007 p54, 55) O RUP e um exemplo de modelo de processo moderno que foi derivado da UML.Ele traz modelo de todos os elementos os modelos genéricos, apóia interação e ilustra boas praticas de especificação de projetos.

RUP reconhece que os modelos convencionais de processo apresentam uma visão única de processos. Existem três perspectivas de descrição.

Dinâmica: mostra as fases do modelo a longo prazo

Estática: mostra as atividades realizadas no processo

Prática: Sugere que a boa pratica a ser usada no processo.

As fases se dividem em:

Concepção: Onde se faz uma avaliação inicial do projeto

Elaboração: Identifica os casos de uso principais do projeto e elabora o software em interações para reorganizar a arquitetura de um sistema, ao final da elaboração deve se tiver uma boa idéia dos requisitos e um esqueleto funcional do sistema, que atue como semente de desenvolvimento.

Construção: Continua o processo, desenvolvendo funcionalidade suficiente para o lançamento

Transição: Inclui varias atividades de ultimo estagio que não são feitas de forma interativa. Isso pode incluir a distribuição pra o cento de dados, treinamento dos usuários e coisas parecidas.

De acordo com as disciplinas no RUP se dividem em seis que são:

Requisitos: Visa levantar o que e pedido pelos interessados no sistema transformar estes pedidos em requisitos que devem funcionar no sistema ha ser construído fornecendo os requisitos necessários do que deve ser feito.

Analise e projeto: define o projeto e mostra como vai ser realizado o sistema objetivando as seguintes tarefas. Executar as tarefas e funções especificadas nas descrições de caso de uso, que cumpra as necessidades propostas e quando ocorrer mudanças de requisitos funcionais seja de fácil modificação. O modelo de projeto esta definido em classes de design estruturado em pacotes e componentes de aplicação. Que contem descrições de como o objeto dessas classes colabora para desempenhar os casos de uso.

Implementação: A implementação serve para definir a organização do código, implementar classes e objetos, testar os componentes desenvolvidos como unidades e integrar os resultados produzidos por desenvolvedores de um sistema executável. Na sua realização deve descrever a reutilização de componentes tornando o sistema mais fácil de manutenção e evolução.

Teste: A disciplina teste visa verificar a interação entre objetos, a integração de todos os componentes do software, se os requisitos forma corretamente implementados, identificar e garantir que os defeitos são abordados antes da implantação do software e garantir que todos os defeitos estão corrigidos, reanalisados e fechados.

Implantação: Tem como objetivo e produzir com sucesso lançamento de produtos e entregar o software ao usuário final. Ele inclui varias atividades como produção de releases externos do software, embalagem do software e aplicativos, distribuição e instalação.

Ambiente: tem como principal enfoque as atividades relacionada com configuração do processo para um projeto, descreve as atividades necessárias para as diretrizes de apoio a um projeto. A finalidade das atividades e organizar os processos e as ferramentas que darão suporte para a equipe no desenvolvimento do siste

De acordo com o site wikipedia (Gerência de Projeto: E o [planejamento de projeto](http://pt.wikipedia.org/wiki/Planejamento_de_projeto) no RUP ocorre em dois níveis. Há uma baixa granularidade ou planos de Fase que descreve todo o projeto, e uma série de alta granularidade ou planos de Iteração que descrevem os passos iterativos. Esta disciplina concentra-se principalmente sobre os aspectos importantes de um processo de desenvolvimento iterativo: Gestão de riscos; Planejamento um projeto iterativo através do ciclo de vida e para uma iteração particular; E o processo de acompanhamento de um projeto iterativo, métricas. No entanto, esta disciplina do RUP não tenta cobrir todos os aspectos do gerenciamento de projetos.

**Artefatos:** são produtos de trabalho finais ou intermediários produzidos e usados durante os projetos. Os artefatos são usados para capturar e transmitir informações do projeto. Um artefato pode ser um dos seguintes elementos, um **documento**, como Caso de Negócio ou Documento de Arquitetura de Software, um **modelo**, como o Modelo de Casos de Uso ou o Modelo de Design, um **elemento do modelo**, ou seja, um elemento existente em um modelo, como uma classe ou um subsistema.

Faz-se necessário para o desenvolvimento deste sistema a utilização do RUP tendo em vista que a sua metodologia ajuda não só ter uma idéia de como o sistema ficara bem como a melhora na velocidade de execução do projeto.