**[Rational Unified Process](http://www.tiespecialistas.com.br/tag/rup/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG" \o "RUP) (**[**RUP**](http://www.tiespecialistas.com.br/tag/rup/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG)**)**

O Processo Unificado (PU) surgiu como um processo popular para o desenvolvimento de software visando à construção de sistemas orientados a objetos (o RUP – Rational Unified Process é um refinamento do PU). É um processo iterativo e adaptativo de desenvolvimento e vem ganhando cada vez mais adeptos devido a maneira organizada e consistente que permite conduzir um projeto

fornecendo uma abordagem disciplinada para assumir tarefas e responsabilidades dentro de uma organização de desenvolvimento, cujo objetivo é assegurar a produção de software de alta qualidade dentro de prazos e orçamentos previsíveis (Kruchten 2003, pág. 14). Derivado dos trabalhos sobre [UML](http://www.tiespecialistas.com.br/tag/uml/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG) e do [Processo](http://www.tiespecialistas.com.br/categoria/ti_corporativa/gestao-de-processos-ti_corporativa/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG) Unificado de [Desenvolvimento de Software](http://www.tiespecialistas.com.br/tag/desenvolvimento-de-software/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG), ele traz elementos de todos os modelos genéricos de processo, apoia a iteração e ilustra boas práticas de especificação e projeto (Sommervillie 2007, pág. 54). Ele captura seis das [melhores práticas](http://www.tiespecialistas.com.br/tag/melhores-praticas/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG) no desenvolvimento de software de forma satisfatória para uma grande faixa de projetos e organizações (Krutchten 2003, pág. 14). As melhores práticas abordadas são as seguintes (Sommerville 2007, pág. 56):

1. **Desenvolver o**[**software**](http://www.tiespecialistas.com.br/tag/software/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG)**iterativamente:** planejar os incrementos de software com base nas prioridades do cliente e desenvolver e entregar o mais cedo possível às características de sistema de maior prioridade no processo de desenvolvimento.

**2. Gerenciar**[**Requisitos**](http://www.tiespecialistas.com.br/tag/requisitos/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG)**:** documentar explicitamente os requisitos do cliente e manter acompanhamento das mudanças desses requisitos. Analisar o impacto das mudanças no sistema antes de aceitá-las.

**3. Usar arquiteturas baseadas em componentes:** Estruturar a arquitetura do sistema com componentes, reduzindo a quantidade de software a ser desenvolvido e, consequentemente, reduzir custos e riscos.

**4. Modelar software visualmente:** usar modelos gráficos de UML para apresentar as visões estática e dinâmica do software.

**5. Verificar a**[**qualidade**](http://www.tiespecialistas.com.br/tag/qualidade/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG)**do software:**garantir que o software atenda aos padrões de qualidade da organização.

**6. Controlar as mudanças do software:**gerenciar as mudanças do software, usando um sistema de gerenciamento de mudanças, procedimentos e ferramentas de gerenciamento de configuração.

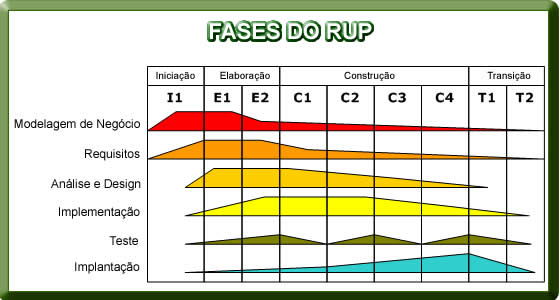
O RUP é um modelo constituído por quatro fases do processo de software, relacionadas mais estritamente aos negócios do que a assuntos técnicos (Sommerville 2007, pág. 54). As quatro fases do RUP são descritas abaixo e representadas graficamente:

**1. Concepção:** o objetivo desta fase é estabelecer um business case para o sistema. Devem ser identificadas todas as entidades externas (pessoas e sistemas) que irão interagir com o sistema em desenvolvimento e definir essas interações. Essas informações são utilizadas para avaliar a contribuição do novo sistema para o negócio.

**2. Elaboração:** os objetivos desta fase são desenvolver um entendimento do domínio do problema, estabelecer um framework de arquitetura para o sistema, desenvolver o plano de projeto e identificar seus principais [riscos](http://www.tiespecialistas.com.br/tag/riscos/?utm_source=site_tag&utm_medium=site&utm_content=09-05-2013&utm_campaign=TAG). Ao final desta fase deve-se ter um modelo de requisitos para o sistema (os casos de uso da UML são especificados), uma descrição de arquitetura e um plano de desenvolvimento do software.

**3. Construção:** está fase está essencialmente relacionada ao projeto, programação e teste do sistema. As partes do sistema são desenvolvidas paralelamente e integradas durante esta fase. Ao final deve-se ter um sistema de software em funcionamento e a documentação associada pronta para ser liberada para os usuários.

**4. Transição:** nesta fase, faz-se a transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários, com a entrada do sistema em funcionamento no ambiente real. Esta é uma atividade ignorada na maioria dos modelos de processo de software, pois é onerosa e às vezes problemática. Ao final desta fase, deve-se ter um sistema de software documentado, funcionando corretamente em seu ambiente operacional.



As atividades que ocorrem durante o processo de desenvolvimento como representadas no gráfico são chamadas de *workflows.* Existem seis *workflows* principais:

|  |  |
| --- | --- |
| **Workflow** | **Descrição** |
| **Modelagem de Negócios** | Os processos de negócio são modelados usando casos de uso de negócios. |
| **Requisitos** | Os agentes que interagem com o sistema são identificados e os casos de uso são desenvolvidos para modelar os requisitos do sistema. |
| **Análise e Projeto** | Um modelo de projeto é criado e documentado usando modelos de arquitetura, modelos de componente, modelos de objetos e modelos de sequência. |
| **Implementação** | Os componentes de sistema são implementados e estruturados em subsistemas de implementação. A geração automática de código com base o modelo de projeto ajuda a acelerar esse processo. |
| **Teste** | O teste é um processo iterativo realizado em conjunto com a implementação. O teste de sistema segue o término da implementação. |
| **Implantação** | Uma versão do produto é criada, distribuída aos usuários e instalada no local de trabalho. |
| **Gerenciamento de Configuração e Mudança** | Este workflow de apoio gerencia mudanças no sistema. |
| **Gerenciamento de Projetos** | Este workflow de apoio gerencia o desenvolvimento do sistema. |
| **Ambiente** | Este workflow está relacionado à disponibilização de ferramentas apropriadas de software para a equipe de desenvolvimento |