

# O Processo Unificado (UP): fases e disciplinas

APRESENTAR, EM DETALHES, AS DUAS DIMENSÕES DO PROCESSO UNIFICADO NA SUA VERSÃO RUP.

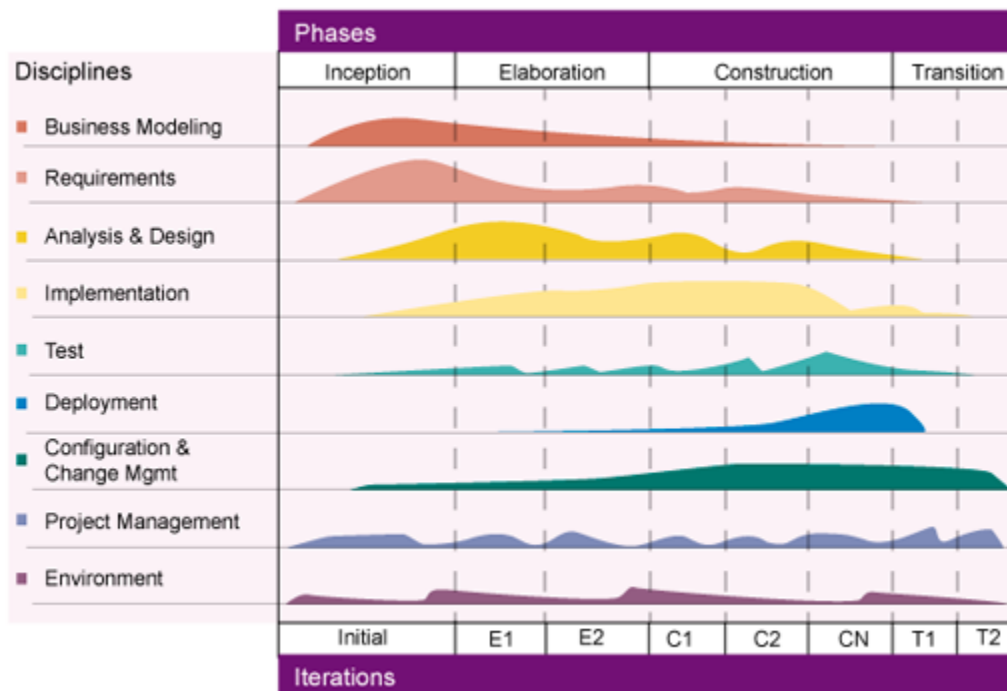
## Visão Geral do RUP

Conforme já comentamos anteriormente, o RUP tem duas dimensões: a dimensão temporal e a dimensão espacial.

Na dimensão temporal (eixo horizontal) temos: Concepção, Elaboração, Construção e Transição.

Já na dimensão espacial (eixo vertical), temos os fluxos, ou workflows: Modelagem de Negócio, Requisitos, Análise e Projeto, Implementação, Teste, Implantação, Gerenciamento de Configuração e Mudança, Gerenciamento de Projeto, Ambiente.

A Figura 1 ilustra os conceitos descritos.



A seguir, vamos ver mais de perto os detalhes de cada um desses componentes.

## Dimensão Temporal (eixo X)

Primeiramente, vamos detalhar o aspecto dinâmico do modelo, com suas várias fases.

### Fase de Iniciação

A finalidade desta fase é definir os objetivos, a viabilidade do projeto e o escopo. Neste período, as seguintes atividades deverão ser executadas:

- Descrever o escopo do projeto.
- Capturar o contexto na forma de requisitos e restrições para determinar um critério de aceitação do produto final.
- Planejar e preparar o Plano de Negócios.
- Elaborar avaliação de riscos, plano de projeto e relações entre custo, cronograma e lucro.
- Preparar uma arquitetura candidata.
- Avaliar alternativas de projeto (essa atividade pode ser suprimida se o sistema não possui novidades ou possui uma arquitetura conhecida).
- Preparar o ambiente de projeto (environment).
- Escolher os recursos físicos e humanos e ferramentas de software.
- Identificar os atores que interagem com o sistema.
- Identificar as interações dos atores com o sistema (casos de uso).

Os seguintes artefatos serão produzidos:

- Documento Visão.
- Modelo de Caso de Uso (inicial).
- Glossário Inicial.
- Plano de Negócios Inicial.
- Análise de Risco Inicial.
- Plano de Projeto (para a etapa de Elaboração).
- Um ou mais protótipos.

### Fase de Elaboração

Nesta fase, deve-se eliminar os elementos de maior risco do projeto através da criação de uma arquitetura coerente e consistente da solução. As seguintes atividades serão executadas:

- Construir protótipos executáveis em uma ou mais interações.
- Construir protótipos evolucionários ou descartáveis, com objetivo de analisar custos-benefícios, demonstrar para investidores, clientes e usuários.

- Assegurar que a arquitetura, os requisitos e os planos sejam estáveis o suficiente e que os riscos sejam suficientemente diminuídos a fim de determinar com segurança o custo e a programação para a conclusão do desenvolvimento.
- Elaborar a visão: entendimento sólido dos casos de uso mais críticos (que determinam as decisões arquiteturais e de planejamento).
- A arquitetura é elaborada e componentes de software são selecionados.
- Componentes potenciais são avaliados segundo decisões fazer/comprar/ reusar para determinar custo e estimativa.
- Lições obtidas podem servir para gerar o novo projeto da arquitetura do sistema.

Os seguintes artefatos serão produzidos:

- Modelo de caso de uso (pelo menos 80% dos casos de uso).
- Especificação de requisitos de software suplementar que capturem requisitos não funcionais e requisitos que não estão associados com um caso de uso específico.
- Descrição da arquitetura de software.
- Protótipo arquitetural executável.
- Uma lista revisada dos riscos e plano de negócios.
- Plano detalhado de desenvolvimento do projeto com interações e critérios de avaliação.
- Um manual do usuário preliminar.

## Fase de Construção

Sua finalidade é desenvolver todos os componentes e características não resolvidas nas fases anteriores, testando-as e integrando-as na forma de um produto, minimizando os custos e construindo versões úteis de forma rápida e prática (alfa, beta e outros releases de teste).

As seguintes atividades deverão ser executadas:

- Decidir se o software, os locais e os usuários estão prontos para que o aplicativo seja implantado.
- Gestão de recursos, controle e otimização do processo.
- Desenvolver e testar os componentes.
- Avaliar o *release* e, eventualmente, prosseguir para a próxima iteração.

Os seguintes artefatos serão produzidos:

- Produto de software integrado na plataforma de hardware.
- Manuais do usuário.
- Descrição dos releases.

## Fase de Transição

Deve garantir que o software esteja disponível para os usuários finais. As seguintes atividades deverão ser executadas:

- "Beta teste".
- Operações paralelas com sistema legado.
- Conversão de bases de dados.
- Treinamento de usuários a mantenedores.
- Roll-out para setores de marketing, distribuição e vendas.
- Ajustar atividades, corrigir erros, etc.
- Finalizar o material de apoio ao usuário final.
- Testar o produto entregue.
- Simular o ambiente do cliente (se possível) ou instalar o software no cliente.
- Realizar um ajuste fino do produto com base no feedback.
- Entregar o produto final para o usuário.

Os seguintes artefatos serão produzidos:

- *Release Notes*.
- É raro o produto que não possui instruções e modificações de "último-minuto".
- Material de treinamento e documentação.

## Dimensão Espacial (eixo Y)

Agora vamos detalhar o aspecto estático do modelo, com seus vários fluxos (ou disciplinas).

## Fluxo de Modelagem de Negócio

Finalidades:

- Documentar processos de negócio usando casos de uso de negócios, com o objetivo de facilitar a comunicação entre as equipes de engenharia de software e a engenharia de negócios.
- Entender a estrutura e a dinâmica da organização na qual um sistema deve ser implantado.
- Entender os problemas atuais da organização-alvo e identificar as possibilidades de melhoria.
- Assegurar que os clientes, usuários e desenvolvedores tenham um entendimento comum da organização.

## Fluxo de Requisitos

Finalidades:

- Descrever o que o sistema deve fazer, de modo que clientes e desenvolvedores concordem sobre esta definição.
- Elicitar, organizar e documentar funcionalidades e restrições.
- Rastrear e documentar compromissos e decisões.
- Oferecer aos desenvolvedores do sistema uma compreensão melhor dos requisitos do sistema.
- Definir as fronteiras do sistema (ou delimitar o sistema).
- Fornecer uma base para planejar o conteúdo técnico das iterações.
- Fornecer uma base para estimar o custo e o tempo de desenvolvimento.

## Fluxo de Análise e Projeto

Finalidades:

- Mostrar como o sistema será concretizado na fase de implementação.
- Provar que o sistema:
  - Executará as tarefas e funções projetadas.
  - Satisfará os requisitos estabelecidos.
  - Será robusto e ameno a mudanças.

## Fluxo de Implementação

Finalidades:

- Organizar o código em subsistemas, camadas, componentes, pacotes.
- Implementar classes e objetos em termos de componentes.
- Testar unitariamente os componentes desenvolvidos.
- Integrar resultados, produzindo um sistema executável.

## Fluxo de Teste

Finalidades:

- Verificar a interação entre objetos.
- Verificar a integração adequada entre os componentes de software.
- Verificar a satisfação dos requisitos.
- Identificar e corrigir defeitos, antes da entrega do software.

## Fluxo de Implantação

Finalidades:

- Realizar entrega bem sucedida do software ao seu cliente, através da:
  - Produção de releases externas.
  - Empacotamento do software.
  - Distribuição do software.
  - Instalação do software.
  - Auxílio aos usuários.

## Fluxo de Gerência de Configuração e Mudança

Tem como finalidade controlar os numerosos artefatos produzidos, ajudando a evitar confusão para garantir que não haverá conflitos no software em decorrência de:

- Atualizações simultâneas.
- Notificação limitada.
- Múltiplas versões.

## Fluxo de Gerência de Projeto

Finalidades:

- Balancear objetivos conflitantes dos envolvidos, superando problemas e entregando, de forma bem sucedida, um produto que satisfaz a necessidade de clientes e usuários.
- Fornecer um suporte para gerenciar projetos de software e gerenciamento de risco.
- Fornecer diretrizes práticas para planejar, montar a equipe, executar e monitorar os projetos.

## Fluxo de Gerência de Ambiente

Tem como finalidade fornecer e garantir o ambiente adequado para a organização, através de ferramentas e processos capazes de suportar as atividades da equipe de desenvolvimento.

Agora que você já estudou esta aula, resolva os exercícios e verifique seu conhecimento. Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

EXERCÍCIOS ([https://ead.uninove.br/ead/disciplinas/impressos/\\_g/pdsoft80\\_100/a14ex01\\_pdsoft80\\_100.pdf](https://ead.uninove.br/ead/disciplinas/impressos/_g/pdsoft80_100/a14ex01_pdsoft80_100.pdf))

## REFERÊNCIA

PRESSMAN, R. S. *Engenharia de Software*. 7. ed. São Paulo: McGraw-Hill, 2010.

SOMMERVILLE, Ian. *Engenharia de Software*. São Paulo: Addison-Wesley, 2007.







