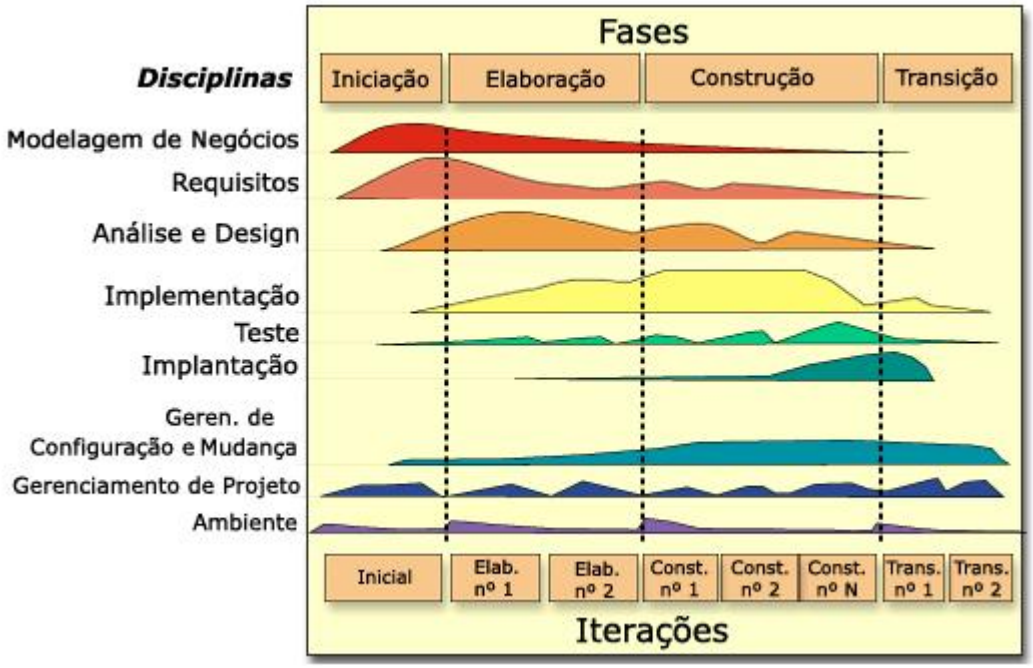
ENGENHARIA DE SOFTWARE

RUP (Rational Unified Process)



O RUP é um **framework** (uma estrutura, um guia mais flexível que uma metodologia rígida) para organizar o desenvolvimento de software.

- **Características Principais:**
  - **Iterativo e Incremental:** O software é construído em ciclos (iterações), e a cada ciclo, uma nova parte funcional é adicionada (incremento).
  - **Centrado na Arquitetura:** A estrutura geral do sistema é definida e validada cedo.
  - **Dirigido por Casos de Uso:** O desenvolvimento é guiado pelas funcionalidades que o usuário realizará (use cases).
  - **Focado em Riscos:** Busca identificar e tratar os maiores riscos no início.

Fases do RUP

O RUP tem 4 fases principais, que marcam momentos importantes do projeto:

1. **Concepção (Inception):**
    - **Objetivo:** Entender a ideia geral do projeto, o escopo básico e se ele é viável. O quê?
    - **Marco (Milestone):** Acordo sobre os objetivos e escopo.
2. **Elaboração (Elaboration):**
    - **Objetivo:** Detalhar os requisitos, definir a arquitetura base do sistema, identificar e mitigar os principais riscos. Como (Estrutura)?
    - **Marco:** Arquitetura base estável e riscos controlados.
3. **Construção (Construction):**
    - **Objetivo:** Construir o software, desenvolvendo o código e integrando as partes. Fazer!
    - **Marco:** Software funcional, pronto para testes finais e entrega inicial.
4. **Transição (Transition):**
    - **Objetivo:** Entregar o software para os usuários finais, fazer ajustes, treinar. Entregar!
    - **Marco:** Sistema entregue e aceito pelos usuários.

Importante: Os **Marcos** estão ligados ao final das **Fases**, indicando que um objetivo importante foi atingido, permitindo decidir se continua ou não.

Disciplinas do RUP

Dentro das fases, várias atividades acontecem, agrupadas em **Disciplinas** (áreas de trabalho/foco):

- **Disciplinas Principais (Engenharia):**
    - **Modelagem de Negócios:** Entender o contexto do negócio onde o software vai operar.
    - **Requisitos:** Levantar e documentar detalhadamente o que o sistema deve fazer (Casos de Uso).

    - **Análise e Design:** Definir a arquitetura e como o sistema será construído tecnicamente (classes, componentes). Transforma requisitos em um projeto técnico.
    - **Implementação:** Escrever e organizar o código. Baseia-se no que foi definido em Requisitos e Análise/Design.
    - **Teste:** Verificar a qualidade, encontrar e corrigir erros.
    - **Implantação:** Instalar e colocar o software em funcionamento para os usuários.
- **Disciplinas de Apoio (Gerenciamento):**
    - **Gerenciamento de Configuração e Mudança:** Controlar versões do código e mudanças nos requisitos.
    - **Gerenciamento de Projetos:** Planejar e acompanhar o projeto (tempo, custo, riscos).
    - **Ambiente:** Preparar as ferramentas e infraestrutura para a equipe trabalhar.

Importante: Uma **Disciplina** não é um produto de trabalho (isso seria um Artefato). Uma disciplina é um conjunto de atividades relacionadas a uma área. O RUP é um processo/framework, enquanto a UML[1] é uma linguagem de modelagem visual que o RUP utiliza para criar diagramas (artefatos).

---

1. **UML (Unified Modeling Language):** Uma linguagem padrão de desenho (visual) para representar as diferentes partes e o funcionamento de um sistema de software. É usada por muitas metodologias, incluindo o RUP, para criar modelos (plantas) do software. **UML é uma linguagem, não uma metodologia.** ↩