

Modelo de métodos formais

Os **métodos formais** são técnicas baseadas em **formalismos matemáticos** para especificação, desenvolvimento e verificação de software, garantindo **confiabilidade e robustez** por meio de análises matemáticas.

Características e Aplicações:

- Usados em sistemas de **alta integridade**, como missões espaciais e controle de voos, devido ao alto custo.
- **Engenharia de software cleanroom**: foca na construção de soluções matematicamente verificáveis antes da implementação.
- Permitem **provas matemáticas**, simulações e garantem a correção do código gerado.
- Métodos notáveis: **Método Z** (amplamente aplicado) e **Gramáticas de Grafos** (para sistemas concorrentes).

Vantagens e Desvantagens:

- ✓ Elimina problemas como **ambiguidade, incompletude e inconsistência**.
- ✓ Pode produzir software **livre de defeitos**.
- ✗ Alto custo e tempo de desenvolvimento.
- ✗ Poucos desenvolvedores possuem a qualificação necessária.
- ✗ Dificil comunicação com clientes devido à complexidade matemática.

Embora eficazes, os métodos formais são aplicáveis apenas a **projetos críticos** devido à sua complexidade e custo elevado.

Processo unificado

O **Processo Unificado (PU)** é um processo de desenvolvimento de software voltado para **sistemas orientados a objetos**, sendo **iterativo e adaptativo**. O **RUP (Rational Unified Process)** é um refinamento dele, trazendo ainda mais estrutura ao processo.

Características do PU:

- **Evolucionário**: baseado em refinamentos sucessivos.
- **Iterativo**: cada iteração funciona como um **miniprojeto**, incluindo análise de requisitos, projeto, implementação e testes.

- **Feedback contínuo:** melhorias são feitas com base na experiência obtida nas iterações anteriores e no retorno dos usuários.
- **Modelado com SPEM (Software Process Engineering Model)**, utilizando **UML** para padronização.

Estruturas do PU:

📌 **Estrutura dinâmica:** representa o **tempo**, sendo dividida em **fases**.

📌 **Estrutura estática:** representa os **elementos do processo**, organizados em **disciplinas**.

O **PU** tem ganhado **popularidade** por oferecer uma abordagem **organizada e consistente** na condução de projetos de software.

Fases

O **Processo Unificado (PU)** organiza o desenvolvimento de software em **quatro fases principais**, cada uma marcada por **maiores marcas (major milestones)**, que servem para revisar e validar o progresso antes de avançar para a próxima etapa.

As Quatro Fases do PU:

- ◆ **1. Concepção (ou iniciação)**
 - Definição inicial e genérica do **escopo do projeto**.
 - Estimativa vaga de **esforço e prazos**.
 - Avaliação de **viabilidade** antes de seguir para a próxima fase.
- ◆ **2. Elaboração**
 - Levantamento detalhado dos **requisitos** (pelo menos 90% ao final).
 - Foco em **riscos e arquitetura**.
 - Implementação inicial dos requisitos mais **críticos**, servindo como base para ajustes e planejamento.
- ◆ **3. Construção**
 - Desenvolvimento iterativo dos **elementos restantes**.
 - Implementação das **funcionalidades finais**.
 - Preparação para **implantação**.
- ◆ **4. Transição**
 - **Testes finais** e refinamentos.
 - **Implantação do sistema** no ambiente de produção.

Cada fase é baseada em **iterações**, garantindo um processo **adaptável e incremental**, reduzindo riscos e melhorando a qualidade final do software. 🚀

Disciplinas

O **Processo Unificado (PU)** organiza as atividades do desenvolvimento de software em **disciplinas**, que são coleções de atividades relacionadas a áreas específicas do processo.

Disciplinas do Processo Unificado

♦ Modelagem de negócios 📊

- Entender a estrutura e dinâmica da organização.
- Identificar problemas e oportunidades de melhoria.
- Alinhar entendimento entre **clientes, usuários e desenvolvedores**.

♦ Requisitos 📝

- Definir **o que o sistema deve fazer**.
- Estabelecer consenso entre **clientes e stakeholders**.
- Fornecer base para planejamento de **custos e prazos**.

♦ Análise e design 🎨

- Transformar requisitos em um **projeto arquitetural**.
- Criar uma estrutura **robusta e escalável**.
- Adaptar o design ao ambiente de **implementação**.

♦ Implementação 💻

- Organizar o código em **subsistemas**.
- Criar e testar **componentes individuais**.
- Integrar tudo em um **sistema executável**.



♦ Teste ✅

- Verificar **interação entre componentes**.
- Garantir que todos os **requisitos foram atendidos**.
- Detectar e corrigir **defeitos** antes da entrega.

♦ Implantação 🚀

- Preparar o software para ser **disponibilizado ao usuário final**.
- Assegurar uma **transição suave** para produção.

♦ Gerenciamento de configuração e mudança 🔄

- Controlar e auditar **modificações no sistema**.
- Definir processos para **gestão de mudanças**.
- ♦ **Gerenciamento de projeto**  17
 - Planejar e monitorar **execução e riscos**.
 - Recrutar e organizar **equipes de desenvolvimento**.
- ♦ **Ambiente** 
 - Definir **ferramentas e processos** que suportam o desenvolvimento.

Características do Processo Unificado

- ✓ **Framework genérico**, adaptável a diferentes projetos.
- ✓ **Baseado em componentes** reutilizáveis.
- ✓ **Utiliza UML** como linguagem de modelagem.
- ✓ **Dirigido por casos de uso**, centrado na **arquitetura**.
- ✓ **Iterativo e incremental**, permitindo ajustes contínuos.
- ✓ **Ágil e realista**, sem planejamento rígido como no modelo cascata.

Em resumo, o **PU** permite desenvolver software de forma **adaptativa**, repetindo atividades conforme necessário e ajustando o planejamento a cada iteração