**Engenharia de Software - Cap. 2**

**Processos de Software**

* **O que é?**

É um conjunto de atividades relacionadas que levam à produção de um produto de software.

* **Quatro atividades fundamentais neste processo:**

1. Especificação, funcionalidades e restrições do software devem ser definidas.
2. Implementação, produção do software para atender às especificações.
3. Validação, verificar se o software atende às demandas do cliente.
4. Evolução, evoluir conforme as necessidades do cliente.

* **Modelos de processo de Software:**

Os modelos apresentados não são mutuamente exclusivos e muitas vezes são usados em conjunto, especialmente em projetos de grande porte.

1. **Cascata**

O estágio seguinte só pode ser iniciado se o anterior foi concluído.

Este modelo não é muito utilizado, pois precisa estabelecer explicitamente todas as necessidades do produto, e o cliente raramente estabelece isso, já que ao decorrer do desenvolvimento o cliente pode decidir novas funcionalidades que deseja em seu software.

**Estágios do modelo:**

1. **Análise e definição de requisitos**, detalhes e restrições do sistema.
2. **Projeto de sistema e software**, definição da arquitetura geral.
3. **Implementação e teste unitário**, desenvolvimento e um teste para cada unidade, verificando se atende à especificação.
4. **Integração e teste do sistema**, unidades individuais do programa são testadas como um sistema completo para assegurar que os requisitos foram atendidos.
5. **Operação e manutenção**, envolve correção de erros não descobertos, melhoras em implementações do sistema.
6. **Incremental**

Desenvolve uma implementação inicial, expõe para o cliente e são criadas várias versões até que o sistema seja adequado às necessidades do cliente.

**Vantagens em relação ao modelo de cascata:**

- A quantidade de documentação é reduzida.

- Fácil de obter o feedback do cliente sobre o desenvolvimento que foi feito.

- É possível obter entrega e implementação rápida do software, já que o cliente pode se utilizar da implementação inicial enquanto as outras implementações estão sendo desenvolvidas.

**Desvantagens:**

-O processo não é visível, não é viável documentar cada versão do sistema.

-A estrutura do sistema tende a se degradar com a adição de novos incrementos.

Portanto este modelo é crítico para sistemas de vida longa, grandes e complexos.

1. **Engenharia de software orientada a reuso**

Utilização de um código de software já existente sabendo que o projeto é semelhante, efetuando ao longo do processo modificações necessárias para se adequar ao projeto.

**Estágios do modelo:**

* 1. **Análise de componentes**, dada as especificações, é efetuada uma busca por componentes para implementar.
  2. **Modificação de requisitos**, efetuada as modificações de acordo com os componentes disponíveis.
  3. **Projeto do sistema com reuso**, o sistema é projetado ou algo inexistente é reusado.
  4. **Desenvolvimento e integração**, softwares e funcionalidades que não podem ser adquiridos externamente serão desenvolvidos e integrados com o sistema principal.

**Tipos de software que podem ser usados em processos de reuso:**

1. Web services desenvolvidos de acordo com padrões de serviço e disponíveis para invocação remota.
2. Coleções de objetos que são desenvolvidas como um pacote a ser integrado com um framework de componentes.
3. Sistemas se software stand-alone configurados para uso em ambiente particular.

* **Especificação de software**

É um processo de definição dos requisitos e restrições relativas ao sistema à ser desenvolvido.

É um processo crítico, pois se ocorrerem erros nesta fase inevitavelmente irá gerar problemas no projeto e na implementação do sistema.

* **Projeto e implementação de software**

É um processo em que ocorre a conversão das especificações do sistema em um sistema executável, nesta fase ocorre uma descrição da estrutura do software a ser implementado.

Projeto de arquitetura, interface, componentes e banco de dados são exemplos de atividades que podem fazer parte deste processo.

* **Validação de software**

É um processo que irá verificar se o software se adequa às suas especificações e as do cliente.

A principal técnica de validação é o teste de programa, em que o sistema é executado com dados de testes simulados.

* **Evolução do software**

É um processo que ocorre para atender às novas necessidades do cliente, sendo assim, é efetuada alterações no software existente.

**O software deve evoluir para continuar útil**

* **Lidando com mudanças**

O projeto deve incluir atividades para lidar com mudanças.

Os processos do projeto podem envolver uma fase de prototipação, que ajuda a evitar más decisões sobre os requisitos do projeto, também podem ser estruturados para o desenvolvimento e entrega iterativos, de forma que mudanças possam ser feitas sem que se afete o sistema como um todo.

* **Rational Unified Process (RUP)**

É um modelo genérico de processo, organizado em fases (concepção, elaboração, construção e transição), mas que separa as atividades (requisitos, análise, projeto, etc) dessas fases.