PROCESSOS DE SOFTWARE

Somerville – capítulo 4
Pressman – pg 30 a 46

1

1

PROCESSO

Um processo é uma série de passos distintos para a produção de um produto ou serviço (P/S)

Pode ser visto como uma cadeia de valor: cada passo adiciona um valor, contribuindo para a criação e entrega do P/S

2

O QUE É PROCESSO DE SOFTWARE?

- É o conjunto de todas as atividades relacionadas ao desenvolvimento e entrega de um software
- Os processos de software ainda são muitos dependentes das pessoas que os executam.
- Dificuldade de automação dos processos de software.
- Buscar primeiro a padronização de processos
- As atividades genéricas em todos os processos de software são:
  - Especificação – o que o sistema deve fazer e suas restrições de desenvolvimento.
  - Projeto – projetar uma estrutura de software que atenda à especificação
  - Desenvolvimento – produção do sistema de software.
  - Validação – verificar se o software faz o que o cliente deseja.
  - Evolução – mudança do software em resposta às demandas de mudança.

3

3

PROCESSOS DE SOFTWARE

6

PROCESSOS DE SOFTWARE

FASE DE DEFINIÇÃO: *"o que"* será desenvolvido.

- Quais informações serão processadas e armazenadas
- Quais as funções e quais as interfaces do SI
- Quais as restrições e quais os critérios de validação
- Qual o desempenho esperado

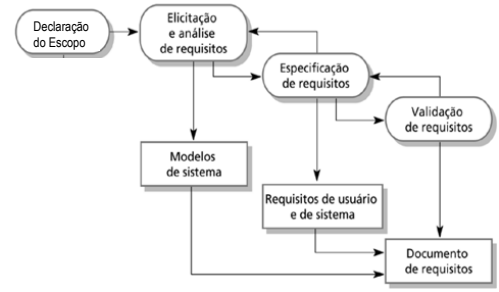
ETAPAS DA FASE DE DEFINIÇÃO

- DEFINIÇÃO DO ESCOPO:
  - identificam as metas gerais do projeto e as funções primárias que o sw deve realizar.
  - Define a abrangência do projeto e delimita estas funções(o que não faz)
  - Documenta as restrições (prazos, $, recursos, tecnologia, interfaces)
  - Não define como atingir.

7

7

O PROCESSO DE ENGENHARIA DE REQUISITOS



8

8

PROCESSOS DE SOFTWARE

DESENVOLVIMENTO: *"como"* o software vai ser desenvolvido.

- Como estruturar os dados
- Como implementar as funções
- Como Testar

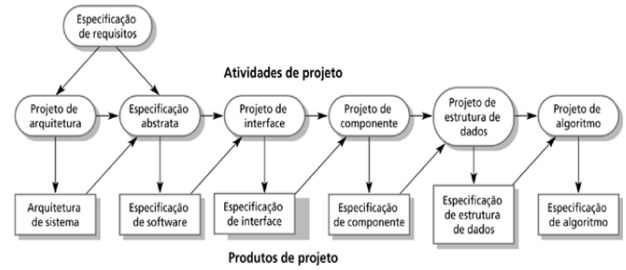
ETAPAS DA FASE DE DESENVOLVIMENTO

- *Projeto de Software*: traduz os requisitos do software num conjunto de representações (algumas gráficas, outras tabulares ou baseadas em linguagem) que descrevem a estrutura de dados, a arquitetura do software, os procedimentos algorítmicos e as características de interface.
- *Codificação*: as representações do projeto devem ser convertidas numa linguagem artificial (a linguagem pode ser uma linguagem de programação convencional ou uma linguagem não procedimental) que resulte em instruções que possam ser executadas pelo computador.
- *Realização de Testes do Software*: logo que o software é implementado numa forma executável por máquina, ele deve ser testado para que se possa descobrir defeitos de função, lógica e implementação.

9

9

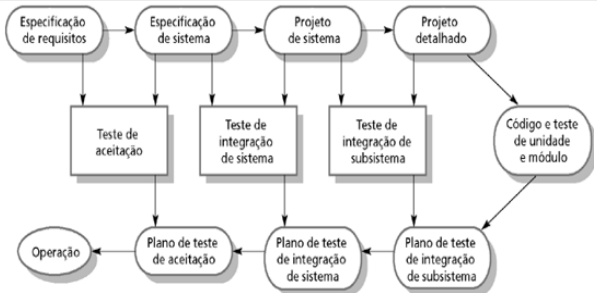
MODELO GERAL DO PROJETO DE SOFTWARE



10

10

MODELO GERAL DE CODIFICAÇÃO E TESTES



11

PROCESSOS DE SOFTWARE

FASE DE MANUTENÇÃO: concentra-se nas "mudanças" que ocorrerão depois que o software for liberado para uso operacional

Correção: consiste da atividade de correção de erros observados durante a operação do sistema – *Manutenção Corretiva*

Adaptação: realiza alterações no software para que ele possa ser executado sobre um novo ambiente (nova arquitetura, novos dispositivos de hardware, novo sistema operacional, novo SGBD...) – *Manutenção Adaptativa*

Melhoramento Funcional: são realizadas alterações para melhorar alguns aspectos do software, como por exemplo, seu desempenho, sua interface, introdução de novas funções, etc... – *Manutenção Perfectiva*

12

12

PROCESSOS DE SOFTWARE

A manutenção do software envolve

- análise do sistema existente (entendimento do código e dos documentos associados),
- Projeto da implementação
- Codificação
- teste das mudanças, teste das partes já existentes, testes de regressão

13

13

CICLO DE VIDA

- **Partes que compõe a vida de um objeto em estudo**

Como todo **produto** industrial, o produto de software tem seu **ciclo de vida:**

- Ele é **concebido** para tentar atender a uma necessidade;
- É **especificado**, quando essas necessidades são traduzidas em requisitos viáveis;
- É **desenvolvido**, transformando-se em um conjunto formado por código e outros itens, como modelos, documentos e dados;
- Passa por algum procedimento de **aceitação** e é entregue a um cliente;
- Entra em **operação**, é usado, e sofre atividades de manutenção, quando necessário;
- É **retirado** de operação ao final de sua vida útil.

- Os Processos de Software definem a metodologia (Paradigma) de desenvolvimento de software, e especificam como executar estas atividades

14

14

MODELOS GENÉRICOS DE PROCESSO DE SOFTWARE

Conjunto de *etapas* que envolve métodos, ferramentas e procedimentos. Alguns paradigmas mais conhecidos são:

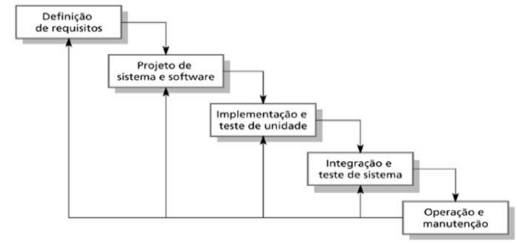
- Ciclo de vida clássico (modelo cascata),
  - Fases separadas e distintas executadas de maneira sequencial, com pequena interação entre fases
- Análise e Projeto Estruturado (décadas 1970 – 1990);
  - Análise Estruturada e Análise Estruturada Moderna (ou Essencial);
- Modelos Iterativos (Desenvolvimento espiral / evolucionário)
  - Especificação, desenvolvimento e validação são intercalados.
- Análise e Projeto Orientado a Objetos (meados 1980 – atual) - RUP;
- Engenharia de software baseada em componentes
  - O sistema é montado a partir de componentes existentes.
- Metodologias Ágeis (XP, SCRUM, KANBAN, LEAN)

15

15

MODELO CASCATA

No desenvolvimento em cascata, o software é construído seguindo uma sequência de fases, sendo que **cada fase**, com exceção da primeira, **depende da conclusão da fase anterior para ser iniciada**.



16

16

Problemas do modelo cascata

- A principal desvantagem do modelo cascata é a dificuldade de acomodação das mudanças depois que o processo está em andamento.
- Apropriado quando os requisitos são bem compreendidos, e quando as mudanças forem bastante limitadas durante o desenvolvimento do sistema.
- Poucos sistemas de negócio têm requisitos estáveis.
- O modelo cascata é o mais usado em projetos de engenharia de sistemas de grande porte, onde um sistema é desenvolvido em várias localidades.
- Permite definir valores e prazos antes de iniciar o projeto
- Normalmente usado em conjunto com os demais modelos

17

17

DESENVOLVIMENTO EVOLUCIONÁRIO

- Desenvolvimento de uma implementação inicial, expondo o resultado aos comentários do usuário e refinando esse resultado por meio de várias "versões" até chegar a um sistema adequado.
- A especificação pode ser desenvolvida de forma incremental
- Exige uma maior participação do usuário
- Projetos são mais difíceis de serem gerenciados
  - Impossível fazer estimativas globais(não existe escopo global)
  - Contratar por tarefa ou por hora trabalhada.

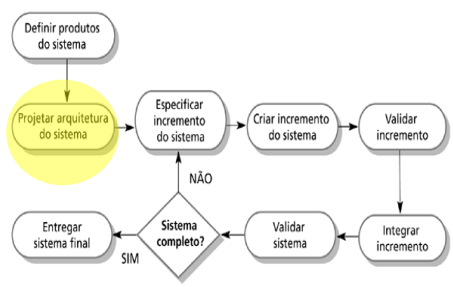
18

18

DESENVOLVIMENTO EVOLUCIONÁRIO

Figura 17.1

Processo de desenvolvimento iterativo.



19

DESENVOLVIMENTO EVOLUCIONÁRIO

- Problemas
  - Falta de visibilidade de processo;
  - Os sistemas são freqüentemente mal estruturados;
- Aplicabilidade
  - Para sistemas interativos de pequeno e médio portes;
  - Para partes de um sistema de grande porte (por exemplo, a interface de usuário);
  - Para sistema com curto ciclo de vida.

20

20

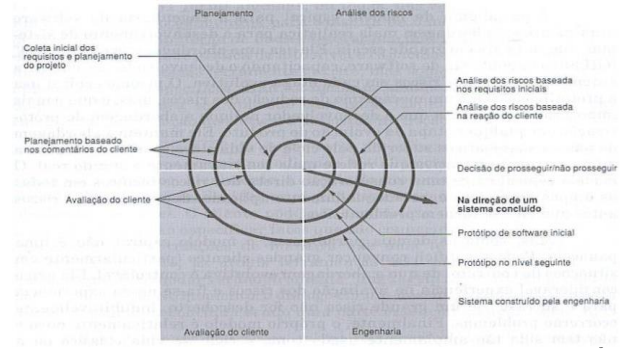
DESENVOLVIMENTO ESPIRAL

- O processo é representado como uma espiral ao invés de uma seqüência de atividades com realimentação.
- Cada *loop* na espiral representa uma fase no processo.
- Sem fases definidas, tais como especificação ou projeto – os *loops* na espiral são escolhidos dependendo do que é requisitado.
- Os riscos são explicitamente avaliados e resolvidos ao longo do processo.

22

22

DESENVOLVIMENTO ESPIRAL



23

ENGENHARIA DE SOFTWARE BASEADA EM COMPONENTES

- Baseado em reuso sistemático onde sistemas são integrados a partir de componentes existentes
- Atividades do processo
  - Análise de componentes;
  - Modificação de requisitos;
  - Projeto de sistema com reuso;
  - Desenvolvimento e integração.
- Esta abordagem está se tornando cada vez mais usada à medida que padrões de componentes têm surgido.

24

24

O RATIONAL UNIFIED PROCESS - RUP

No desenvolvimento RUP o processo é empregado de forma **iterativa e incremental**, mas ainda existe uma forte linearidade no desenvolvimento, caracterizada por **cascatas menores** dentro de cada iteração.

26

26

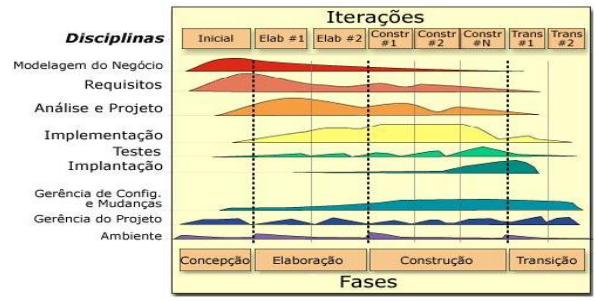
Fases do RUP

- Concepção
  - Estabelecer o *business case* para o Sistema (estudo de viabilidade e modelagem do negócio)
- Elaboração
  - Desenvolver um entendimento do domínio do problema e a arquitetura do sistema.
- Construção
  - Projeto, programação e teste de sistema.
- Transição
  - Implantar o sistema no seu ambiente operacional.

28

28

O RATIONAL UNIFIED PROCESS - RUP



29

MÉTODOS ÁGEIS

- A insatisfação com os overheads envolvidos nos métodos de projeto levou à criação dos métodos ágeis. Esses métodos:
  - Enfocam o código ao invés do projeto;
  - São baseados na abordagem iterativa para desenvolvimento de software;
  - São destinados a entregar software de trabalho e evoluí-lo rapidamente para atender aos requisitos que se alteram.
  - Foco nas pessoas e não nos processos
- Os métodos ágeis são muito usados para sistemas web e de negócio de porte pequeno/médio, mas também para o desenvolvimento evolucionário de sistemas de grande porte

30

30

Métodos ágeis

- **Minimizar o risco** pelo desenvolvimento do software em curtos períodos (chamados de iteração), de 1 semana a 4 semanas.
- **Cada iteração é como um projeto em miniatura** (inclui todas as tarefas necessárias para implantar os mini-incrementos de cada iteração)
- A cada iteração existe a **entrega** de software com **novas funcionalidades**
- O desenvolvimento Ágil enfatiza a **comunicação face-a-face**, por isso produzem **pouca documentação** em relação a outros métodos, sendo este um de seus pontos negativos.
- Eliminar grande parte do **excesso de modelos** e de documentação e o tempo gasto nestas tarefas

31

31

ESCOLHA DE UM PARADIGMA DE SOFTWARE

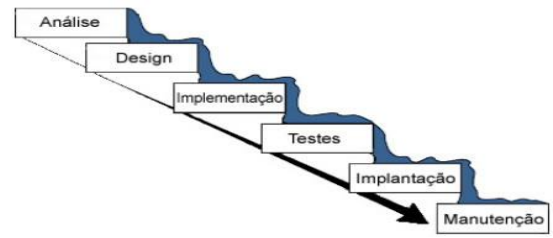
- Existem várias alternativas de solução. Qual a melhor?
- Cada organização (gestor) identificará o conjunto de soluções que melhor se adapte às suas características:
  - Desenvolvimento Interno / Externo
  - Fábrica de software
  - Tipos de Sistemas
  - Tamanho dos projetos
  - Tecnologia disponível
  - Tamanho e Maturidade da Equipe
  - Cultura Empresarial – Estágio de Informatização da organização / mercado para quem são desenvolvidos os produtos

32

32

ESCOLHA DE UM PARADIGMA DE SOFTWARE

O que há em comum em todos os paradigmas?



33

33