

*Universidade Federal do Espírito Santo
Centro Tecnológico
Departamento de Informática*



Disciplina: INF15978 – Engenharia de Software I

Prof.: Monalessa Perini Barcellos

(monalessa@inf.ufes.br)

1

Conteúdo

1. **Introdução à Engenharia de Software**
2. **Processo de Software**

2

1. Introdução à Engenharia de Software

- Erroneamente, muitas vezes, *desenvolver software* é confundido com *programação*.
- Programação é uma abordagem (*abordagem de construção*) corretamente adotada quando se pensa em problemas pequenos. Por exemplo: calcular médias e ordenar conjuntos de dados.
- Para problemas mais complexos é necessário adotar uma *abordagem de engenharia*. Por exemplo: desenvolver um sistema de informações para um banco.

Engenharia de Software

Monalessa Perini Barcellos

3

1. Introdução à Engenharia de Software

Analogamente:



Construir uma casinha de cachorro é um problema simples. O próprio dono pode comprar os materiais e construir a casinha em um final de semana.



Construir um edifício requer um projeto de engenharia civil, planejamento da execução da obra e desenvolvimento de modelos (maquetes e plantas de diversas naturezas), até a realização da obra, que deve ocorrer por etapas. Ao longo da realização do trabalho, é necessário realizar um acompanhamento para verificar prazos, custos e a qualidade do que se está construindo.

Engenharia de Software

Monalessa Perini Barcellos

4

1. Introdução à Engenharia de Software

- A Engenharia de Software surgiu com o objetivo de melhorar a qualidade dos produtos de software e aumentar a produtividade no processo de desenvolvimento.
- Trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software.
- Propõe a divisão do problema em problemas menores, cujas soluções devem ser integradas por uma arquitetura. As soluções devem ser obtidas utilizando-se procedimentos (métodos, técnicas, roteiros etc.), bem como ferramentas que automatizam o trabalho (ou parte dele). Tipicamente, são exigidas várias pessoas, cujo esforço deve ser planejado, coordenado e acompanhado. Também é requerido que a qualidade do que se está produzindo seja sistematicamente avaliada.

Engenharia de Software

Monalessa Perini Barcellos

5

1. Introdução à Engenharia de Software

“A Engenharia de Software surgiu com o objetivo de melhorar a **qualidade dos produtos de software** e aumentar a produtividade no processo de desenvolvimento.”

O que é um produto de software de qualidade?



Desenvolvedor (*perspectiva interna*): “É um produto fácil de manter.”



Usuário (*perspectiva externa de observação pelo uso do produto*): “É um produto que **satisfaz** minhas necessidades, é fácil de usar, eficiente e confiável.”



Cliente (*perspectiva externa de observação da qualidade em uso*): “É um produto que **agrega valor** a meu negócio .”

Engenharia de Software

Monalessa Perini Barcellos

6

1. Introdução à Engenharia de Software

Então:

Qualidade de software é um conceito com múltiplas facetas (perspectivas de usuário, desenvolvedor e cliente) e que envolve diferentes características (por exemplo, *usabilidade, confiabilidade, eficiência, manutenibilidade, portabilidade, segurança, produtividade*) que devem ser alcançadas em níveis diferentes, dependendo do propósito do software.

Mas...

Esse conceito foca produto.

1. Introdução à Engenharia de Software

Então:

Qualidade de software é um conceito com múltiplas facetas (perspectivas de usuário, desenvolvedor e cliente) e que envolve diferentes características (por exemplo, *usabilidade, confiabilidade, eficiência, manutenibilidade, portabilidade, segurança, produtividade*) que devem ser alcançadas em níveis diferentes, dependendo do propósito do software.

Mas...

Esse conceito foca produto.

Como garantir que o produto de software tenha qualidade?

1. Introdução à Engenharia de Software

relaciona-se diretamente com a

Qualidade do Produto de Software

Qualidade do Processo de Software

Melhorando a qualidade do processo de software, é possível melhorar a qualidade dos produtos resultantes.

- A premissa por detrás dessa afirmativa é a de que processos bem estabelecidos, que incorporam mecanismos sistemáticos para acompanhar o desenvolvimento e avaliar a qualidade, no geral, conduzem a produtos de qualidade.
- Exemplo: série ISO 9000.

Engenharia de Software

Monalessa Perini Barcellos

9

2. Processo de Software

De modo geral, o que é um PROCESSO?



Uma receita de bolo é um processo?



A abertura de uma conta em um banco é um processo?

- Um **processo de software** pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software.
- Um processo eficaz deve, claramente, considerar as relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido.

Engenharia de Software

Monalessa Perini Barcellos

10

2. Processo de Software

Elementos que compõem um processo de software:

Processo de Software

Processos

Atividades

Pré-atividades

Subatividades

Artefatos

Insumos

Produtos

Recursos

Recursos Humanos

Ferramentas de Software

Hardware

Procedimentos

Métodos

Técnicas

Roteiros

Exemplo:

Atv1. Realizar levantamento de requisitos.

Pré-Atividade: Atv0

Insumo: Documento Planejamento de Entrevista

Produto: Documento Registro de Entrevista.

Recurso Humano: Analista de Sistemas

Procedimento: Técnica para Realização de Entrevistas

Atv2. Documentar requisitos.

Pré-Atividade: Atv1

Insumo: Documento Registro de Entrevista.

Produto: Especificação Textual de Requisitos.

Recurso Humano: Analista de Sistemas

Procedimento: Roteiro para Elaboração da Especificação Textual de Requisitos.

Engenharia de Software

Monalissa Perini Barcellos

11

2. Processo de Software

Processos de software, em uma abordagem de Engenharia de Software, envolvem diversas atividades que podem ser classificadas quanto ao seu propósito em:

- i. *Atividades de Desenvolvimento (ou Técnicas ou de Construção):* são as atividades diretamente relacionadas ao processo de desenvolvimento do software, ou seja, que contribuem diretamente para o desenvolvimento do produto de software a ser entregue ao cliente. Ex.: especificação e análise de requisitos, projeto e implementação.
- ii. *Atividades de Gerência de Projeto:* são aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto, tais como realização de estimativas, elaboração de cronogramas, análise dos riscos do projeto etc.
- iii. *Atividades de Apoio:* são aquelas relacionadas principalmente com a garantia da qualidade do produto em desenvolvimento e do processo de software utilizado, tais como revisões e inspeções de produtos (intermediários ou finais) do desenvolvimento.

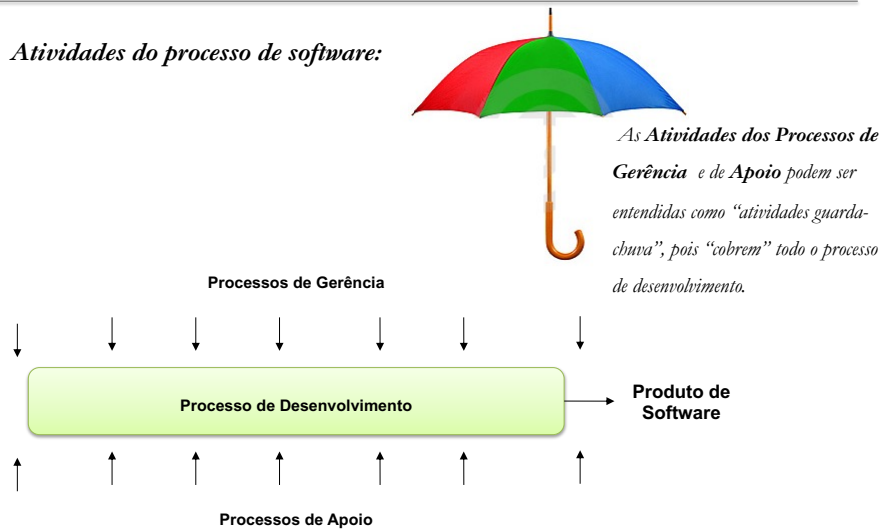
Engenharia de Software

Monalissa Perini Barcellos

12

2. Processo de Software

Atividades do processo de software:



Engenharia de Software

Monalessa Perini Barcellos

13

2. Processo de Software

Definição de Processos de Software



Há várias maneiras diferentes de fazer um bolo de chocolate.

Pessoas alérgicas a leite precisam de receitas que não contenham lactose.

Pessoas diabéticas precisam de receitas que não incluam açúcar.



Há várias formas diferentes de abrir uma conta bancária.

Algumas pessoas podem abrir contas apenas para receberem seus salários.

Algumas pessoas podem abrir contas com vantagens especiais, como limite de crédito.

Processos de software podem ter diversas definições distintas.

Engenharia de Software

Monalessa Perini Barcellos

14

2. Processo de Software

O que deve ser considerado para definir um processo de software?

- Características da aplicação (domínio do problema, tamanho, complexidade etc)
- Tecnologia a ser adotada na sua construção (paradigma de desenvolvimento, linguagem de programação, mecanismo de persistência etc)
- Organização onde o produto será desenvolvido
- Características da equipe
- Estabilidade dos requisitos
- Outros

Modelos de Ciclo de Vida fornecem o arcabouço para a definição do processo.

2. Processo de Software

Modelo de Ciclo de Vida de Software ou Modelo de Processo de Software

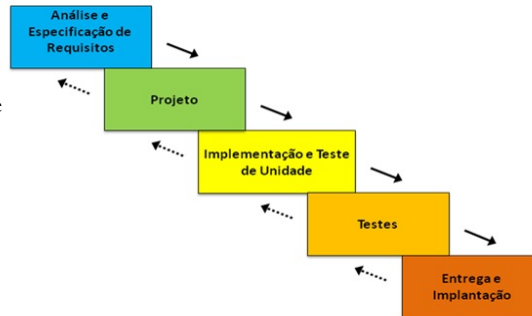
- Representação abstrata de um esqueleto de processo, incluindo tipicamente algumas atividades principais, a ordem de precedência entre elas e, opcionalmente, artefatos requeridos e produzidos.
- É um importante ponto de partida para definir como o projeto deve ser conduzido, mas a sua adoção não é o suficiente para guiar e controlar um projeto de software na prática.
- Geralmente envolve as seguintes fases: *Análise e Especificação de Requisitos*, *Projeto*, *Implementação*, *Testes*, *Entrega e Implantação*, *Operação*, *Manutenção*.
- Os principais modelos de ciclo de vida podem ser agrupados em três categorias principais: *modelos sequenciais*, *modelos incrementais* e *modelos evolutivos*.

2. Processo de Software

Modelos Sequenciais – Modelo Cascata (Modelo de Ciclo de Vida Clássico)

- Uma fase só deve ser iniciada após a conclusão daquela que a precede.
- Uma vez que, na prática, essas fases se sobrepõem de alguma forma, geralmente, permite-se um retorno à fase anterior para a correção de erros encontrados.
- A entrega do sistema completo ocorre somente ao final da fase de Entrega e Implantação.
- O uso de revisões ao fim de cada fase permite o envolvimento do usuário.
- Cada fase serve como uma base aprovada e documentada para o passo seguinte, facilitando bastante a gerência de configuração.

É o modelo de ciclo de vida mais antigo e mais amplamente usado. Indicado para problemas bastante pequenos e bem definidos, onde os desenvolvedores conhecem bem o domínio do problema e os requisitos podem ser claramente estabelecidos.



Engenharia de Software

Monalessa Perini Barcellos

17

2. Processo de Software

Modelos Sequenciais – Modelo Cascata (Modelo de Ciclo de Vida Clássico)

Problemas:

- Projetos reais muitas vezes não seguem o fluxo sequencial que o modelo propõe.
- Os requisitos devem ser estabelecidos de maneira completa, correta e clara logo no início de um projeto. A aplicação deve, portanto, ser entendida pelo desenvolvedor desde o início do projeto.
- O usuário precisa ser paciente. Uma versão operacional do software não estará disponível até o final do projeto.
- A introdução de certos membros da equipe, tais como projetistas e programadores, é frequentemente adiada desnecessariamente. A natureza linear do ciclo de vida clássico leva a “estados de bloqueio” nos quais alguns membros da equipe do projeto precisam esperar que outros membros da equipe completem tarefas dependentes.

Engenharia de Software

Monalessa Perini Barcellos

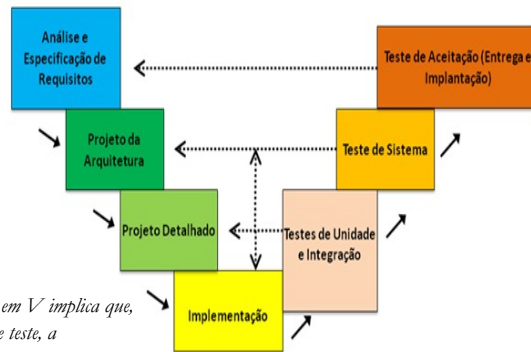
18

2. Processo de Software

Modelos Sequenciais – Modelo em V

- Testes de unidade e integração devem garantir que todos os aspectos do projeto do sistema foram implementados corretamente no código.
- Teste de sistema busca verificar se o sistema atende aos requisitos definidos na especificação.
- Testes de aceitação, conduzidos tipicamente pelos usuários e clientes, validam os requisitos, confirmando que os requisitos corretos foram implementados no sistema.

Varição do Cascata que procura enfatizar a estreita relação entre as atividades de teste (teste de unidade, teste de integração, teste de sistema e teste de aceitação) e as demais fases do processo.



A conexão entre os lados direito e esquerdo do modelo em V implica que, caso sejam encontrados problemas em uma atividade de teste, a correspondente fase do lado esquerdo e suas fases subsequentes podem ter de ser executadas novamente para corrigir ou melhorar esses problemas.

Engenharia de Software

Monalessa Perini Barcellos

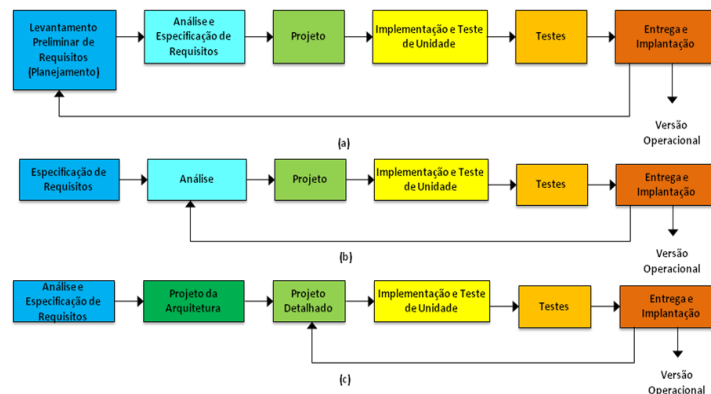
19

2. Processo de Software

Modelos Incrementais – Modelo Incremental

Seu princípio fundamental é que, a cada ciclo ou iteração, uma versão operacional do sistema será produzida e entregue para uso ou avaliação detalhada do cliente.

Variações



Engenharia de Software

Monalessa Perini Barcellos

20

2. Processo de Software

Modelos Incrementais – Modelo Incremental

Vantagens:

- Menor custo e menos tempo são necessários para se entregar a primeira versão;
- Os riscos associados ao desenvolvimento de um incremento são menores, devido ao seu tamanho reduzido;
- O número de mudanças nos requisitos pode diminuir devido ao curto tempo de desenvolvimento de um incremento.

Desvantagens:

- Se os requisitos não são tão estáveis ou completos quanto se esperava, alguns incrementos podem ter de ser bastante alterados;
- A gerência do projeto é mais complexa, sobretudo quando a divisão em subsistemas inicialmente feita não se mostrar boa.

Engenharia de Software

Monalessa Perini Barcellos

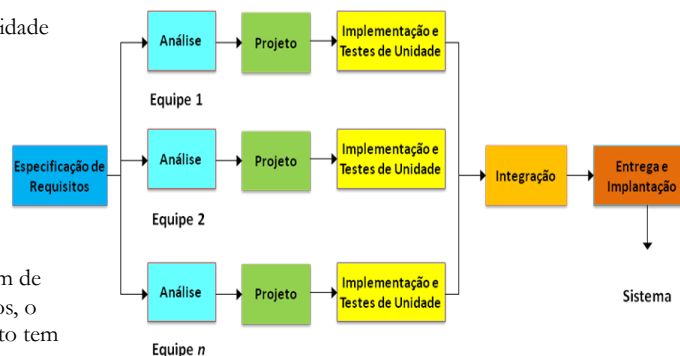
21

2. Processo de Software

Modelos Incrementais – RAD (Rapid Application Development)

Prima por um ciclo de desenvolvimento curto (tipicamente de até 90 dias). Os incrementos são desenvolvidos em paralelo por equipes distintas e apenas uma única entrega é feita.

- Exige disponibilidade de equipes.
- Os requisitos têm de ser bem definidos, o escopo do projeto tem de ser restrito e o sistema modular.



Engenharia de Software

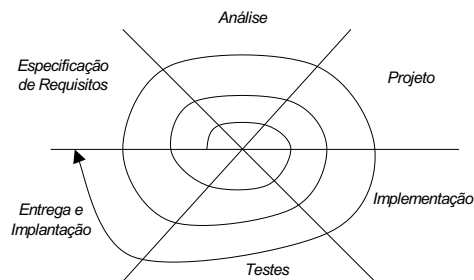
Monalessa Perini Barcellos

22

2. Processo de Software

Modelos Evolutivos – Modelo Espiral

- O sistema é desenvolvido em ciclos, sendo que nos primeiros ciclos nem sempre todas as atividades são realizadas (por exemplo, o produto resultante do primeiro ciclo pode ser uma especificação do produto ou um estudo de viabilidade).
- As passadas subsequentes ao longo da espiral podem ser usadas para desenvolver protótipos, chegando progressivamente a versões operacionais do software, até se obter o produto completo.
- A cada ciclo, o planejamento deve ser revisto com base no feedback do cliente, ajustando, inclusive, o número de iterações planejadas.
- Pode ser difícil convencer clientes, especialmente em situações envolvendo contrato, que a abordagem evolutiva é gerenciável.



Engenharia de Software

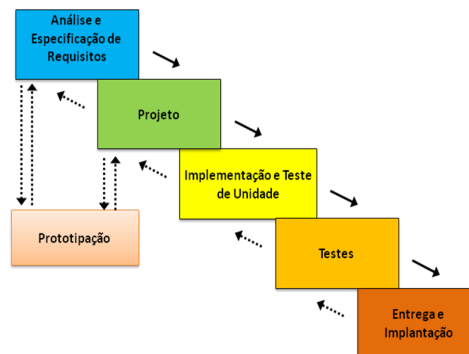
Monalessa Perini Barcellos

23

2. Processo de Software

Prototipação

- Muitas vezes, clientes têm em mente um conjunto geral de objetivos para um sistema de software, mas não são capazes de identificar claramente as funcionalidades ou informações (requisitos) que o sistema terá de prover ou tratar.
- A prototipação é uma técnica para ajudar engenheiros de software e clientes a entender o que está sendo construído quando os requisitos não estão claros.
- Pode ser aplicada no contexto de qualquer modelo de processo (na figura: modelo Cascata com Prototipação).



Engenharia de Software

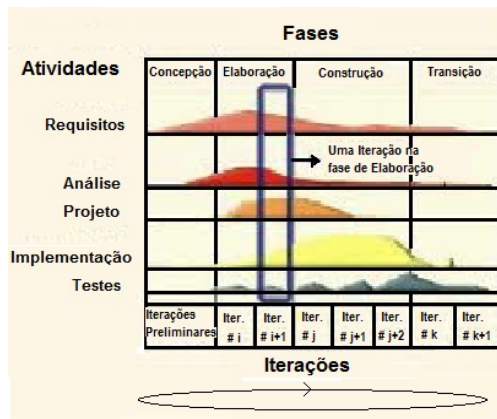
Monalessa Perini Barcellos

24

2. Processo de Software

O Processo Unificado - RUP (Rational Unified Process)

- É um modelo evolutivo que preconiza o desenvolvimento em ciclos, de modo a permitir uma melhor compreensão dos requisitos.
- É mais que um modelo de processo de desenvolvimento. É uma abordagem completa para o desenvolvimento de software, incluindo, além de um modelo de processo de software, a definição detalhada de responsabilidades (papéis), atividades, artefatos e fluxos de trabalho, dentre outros.



Engenharia de Software

Monalessa Perini Barcellos

25

Universidade Federal do Espírito Santo

Centro Tecnológico

Departamento de Informática



Disciplina: INF15978 – Engenharia de Software I

Prof.: Monalessa Perini Barcellos

(monalessa@inf.ufes.br)

26