



Unidade 2 – Processos de Software



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPU SP

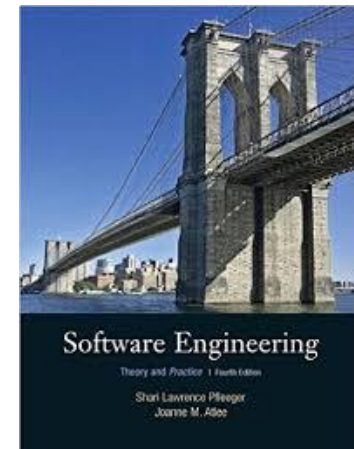
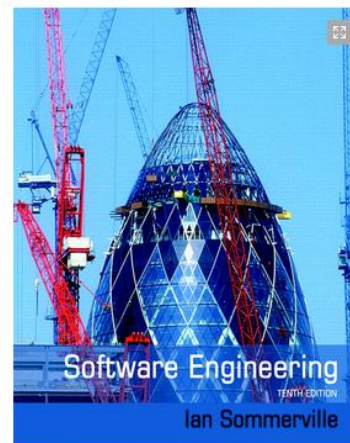


Bibliografia

- **Software Engineering – A Practitioner's Approach – Roger S. Pressman – Eight Edition – 2014**
- **Software Engineering – Ian Sommerville – 10th edition – 2015**
- **Software Engineering – Pfleeger & Atlee – Theory and Practice – 4th edition – Prentice Hall 2009**
- Engenharia de Software – Uma abordagem profissional – Roger Pressman - McGraw Hill, Sétima Edição - 2011
- Engenharia de Software – Ian Sommerville – Nona Edição – Addison Wesley, 2007
- Engenharia de Software – Teoria e Prática - Shari Lawrence Pfleeger – Editora Pearson – 3^a edição



Software Engineering: A
Practitioner's Approach, 8/e





Qual a sistemática empregada pela Engenharia de Software ?





Processos de Software

- A sistemática utilizada pela Engenharia de Software é, algumas vezes, chamada de Processo de Software.
- Um processo de software é uma sequência de atividades que levam à produção de um produto de software.
- Corresponde a uma metodologia para as atividades, ações e tarefas necessárias para se desenvolver software de qualidade.





Processos de Software

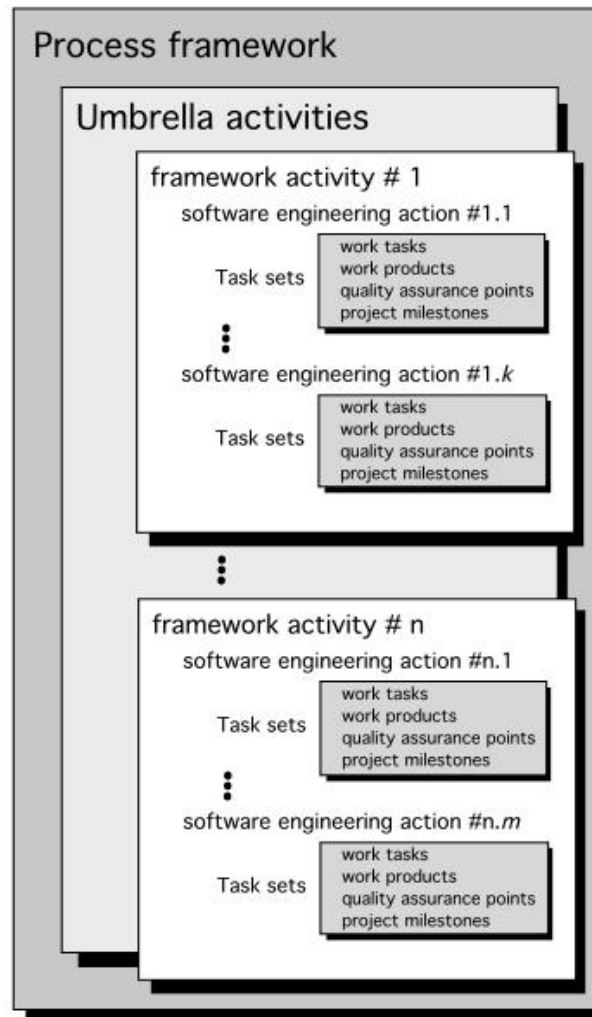
- Uma atividade é um esforço realizado para atingir-se um objetivo amplo (por exemplo, especificação do software)
- Uma ação é um conjunto de tarefas que gera algum artefato de software, por exemplo, a modelagem estrutural do sistema.
- Uma tarefa concentra-se em um objetivo menor, porém bem definido, por exemplo realizar um teste de unidade.





Processos de Software

Software process



Exemplo

- Atividade: Projeto de Software
- Ação: Projeto da Interface
- Tarefa: Protótipo da Interface



Processos de Software Atividades



- **Especificação** de Software, onde os clientes e os engenheiros de software definem a funcionalidade do software que será produzido e as restrições em sua operação.
- **Projeto e Implementação** do Software, onde o software é projetado e programado para atender a especificação.
- **Validação**, onde o software é checado para assegurar que atende aos requisitos do cliente.
- **Evolução**, onde o software é modificado para atender necessidades mutáveis do cliente e do mercado.



Fonte: Sommerville



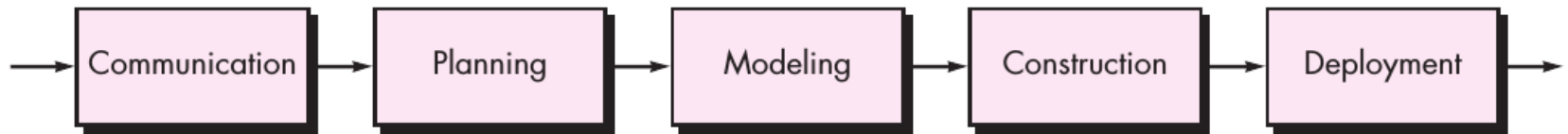
Fluxo de Software

- Descreve como são organizadas as atividades metodológicas, bem como as ações e tarefas que ocorrem dentro de cada atividade em relação à sequência e ao tempo.

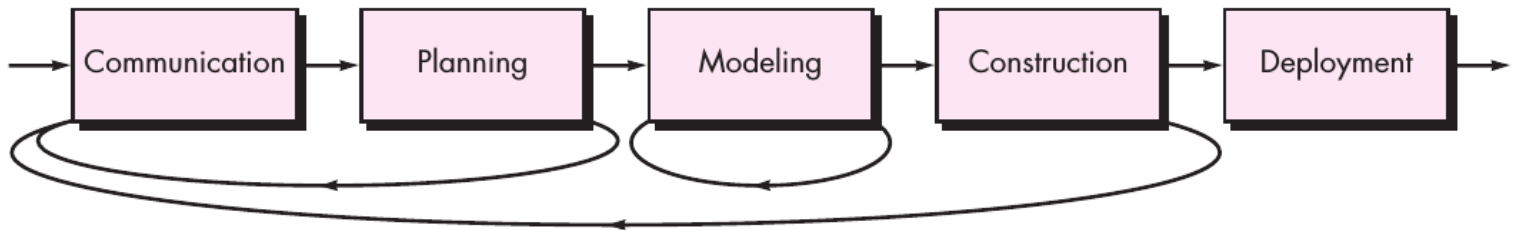




Fluxo de Software



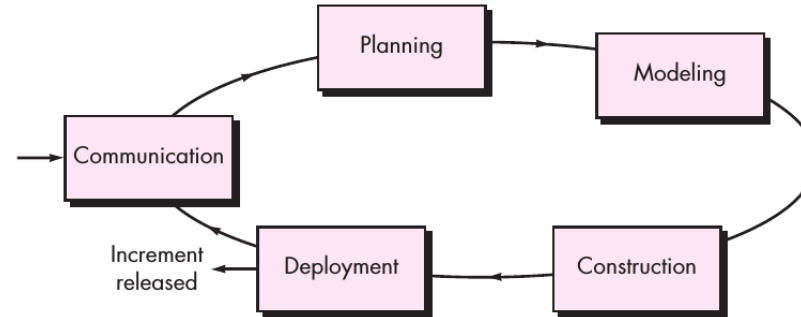
Linear: As atividades são executadas em sequência.



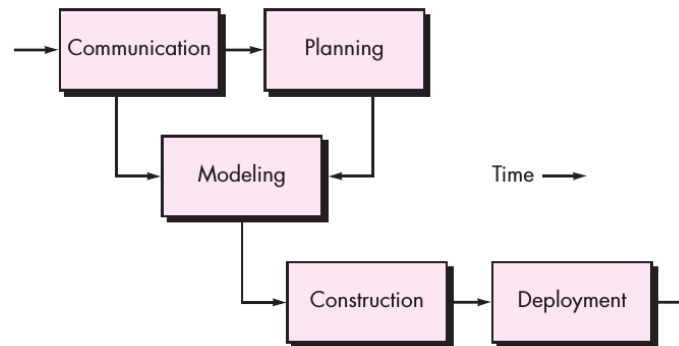
Iterativo: Repete-se uma ou mais atividades antes de proceder à próxima.



Fluxo de Software



Evolucionário: Executa as atividades de modo “circular”. Cada ciclo gera uma versão mais completa do software.



Paralelo: Executa-se uma ou mais atividades em paralelo com outras atividades.



Engenharia de Software

- Corresponde à aplicação de abordagens disciplinadas e sistemáticas para o desenvolvimento, operação e manutenção de software.

software engineering. (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software;

Norma IEEE Standard 610.12 –1990





Processo de Software é sinônimo de Engenharia de Software ?





Processo x Engenharia de Software

- Processo de Software é definido como uma metodologia para as atividades, ações e tarefas necessárias para se desenvolver software com qualidade.
- Mas, a Engenharia de Software também considera tecnologias que fazem parte do processo – Métodos Técnicos e Ferramentas automatizadas.



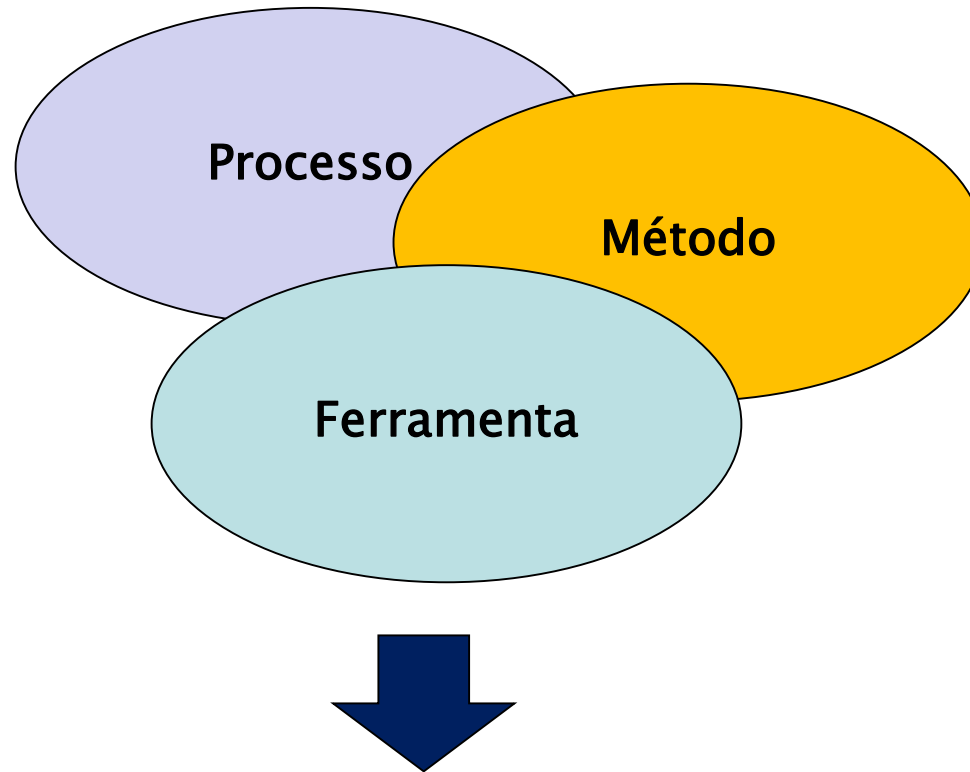
Tecnologia em Camadas



Engenharia de Software



Elementos da Engenharia de Software



Engenharia de Software



Qual a diferença entre processo, método e ferramenta ?





Processo

- Apresenta o passo-a-passo para o desenvolvimento e manutenção do software (conjunto de atividades, framework).
- Corresponde a uma estrutura geral no qual os métodos e as ferramentas são empregadas.





Método

- Descreve como fazer um determinado passo do processo.
- Corresponde aos “Hows-To”.





Ferramentas

- São elementos que tornam mais rápida a execução dos métodos





Qual é o processo ?

1. Coloque o leite em uma panela, junte a farinha de trigo, a margarina, a gema e o sal.
2. Leve ao fogo e mexa bem com um garfo, até soltar a massa da panela.
3. deixe esfriar, faça bolinhas, recheie com um pedacinho de queijo.
4. passe em clara de ovo só misturada e na farinha de rosca.
5. Leve à frigideira e frite em óleo bem quente.





Qual é o processo ?

Processo

1. Coloque o leite em uma panela, junte a farinha de trigo, a margarina, a gema e o sal.
2. Leve ao fogo e mexa bem com um garfo, até soltar a massa da panela.
3. deixe esfriar, faça bolinhas, recheie com um pedacinho de queijo.
4. passe em clara de ovo só misturada e na farinha de rosca.
5. Leve à frigideira e frite em óleo bem quente





Qual é o método ?

1. Coloque o leite em uma panela, junte a farinha de trigo, a margarina, a gema e o sal.
2. Leve ao fogo e mexa bem com um garfo, até soltar a massa da panela.
3. deixe esfriar, faça bolinhas, recheie com um pedacinho de queijo.
4. passe em clara de ovo só misturada e na farinha de rosca.
5. Leve à frigideira e frite em óleo bem quente.





Qual é o método ?

1. **Coloque** o leite em uma panela, **junte** a farinha de trigo, a margarina, a gema e o sal.
2. **Leve** ao fogo e **mexa** bem com um garfo, até soltar a massa da panela.
3. **deixe** esfriar, faça bolinhas, **recheie** com um pedacinho de queijo.
4. **passe** em clara de ovo só misturada e na farinha de rosca.
5. **Leve** à frigideira e **frite** em óleo bem quente





Qual é a **ferramenta** ?

1. Coloque o leite em uma **panela**, junte a farinha de trigo, a margarina, a gema e o sal.
2. Leve ao fogo (no **fogão**) e mexa bem com um **garfo**, até soltar a massa da **panela**.
3. deixe esfriar, faça bolinhas, recheie com um pedacinho de queijo.
4. passe em clara de ovo só misturada e na farinha de rosca.
5. Leve à **frigideira** e frite em óleo bem quente





Método, Técnica e Processo

Processo => Algoritmo

Método => Iterativo, Recursivo

Ferramenta => Eclipse, Netbeans



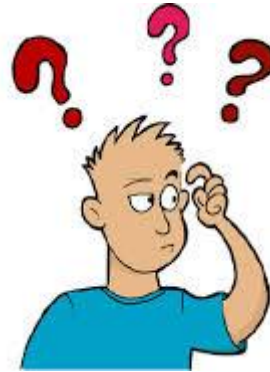
Processos, Métodos e Ferramentas



- ✓ A base para a Engenharia de Software é a camada de processos.
- ✓ Processo é a liga que mantém as camadas coesas e possibilita o desenvolvimento de software de forma racional e dentro do prazo.
- ✓ Os métodos da Engenharia de Software baseiam-se em um conjunto de princípios básicos que definem o procedimento a ser seguido para o processo. Exemplo: Metodologia Estruturada, Metodologia Orientada a Objetos, Método Essencial, etc.
- ✓ Ferramentas fornecem o suporte automatizado para o processo e para os métodos (CASE – Computer Aided Software Engineering).



Há uma método universal que se aplica para todos os tipos de software ?





Em outras palavras, desenvolver um game é o mesmo que desenvolver uma aplicação Cliente Servidor em Mainframe ?





Questões que se aplicam a todos os softwares

- **Não** há um método universal que se aplique para todos os tipos de softwares.
- Mas, existem alguns pontos que são aplicáveis à todos os tipos de softwares.



Princípios que se aplicam a todos os softwares

- Sistemas devem ser desenvolvidos usando um processo de desenvolvimento gerenciado.
- Confiabilidade e desempenho são importantes para todos os tipos de software.
- Especificação do software (o que ele deve fazer) é importante para todo tipo de software.
- Sempre que possível, deve-se **reusar** software existente do que escrever novo software.
- Cada vez mais plataformas heterogêneas são requeridas.
- Volatilidade de Negócios (software deve ser manutenível).





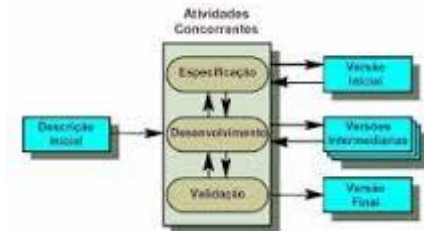
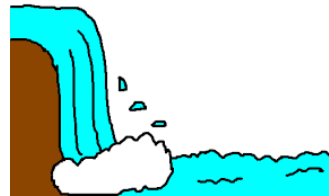
Modelos de Processos de Software

- Correspondem aos paradigmas de desenvolvimento de software.
- Os primeiros Modelos de Processos de Software foram propostos para trazer ordem ao caos existente na área de desenvolvimento de software (Crise de Software).
- Esses modelos tradicionais trouxeram uma importante contribuição à área de Engenharia de Software.
- Mas, software evolui com mudanças e por isso, modelos deve haver um compromisso entre ordem e flexibilidade para se produzir software.



Modelos de Processos de Software

- Modelo Cascata
- Desenvolvimento Incremental
- Desenvolvimento Evolucionário
- Engenharia de Software Orientada a Reuso





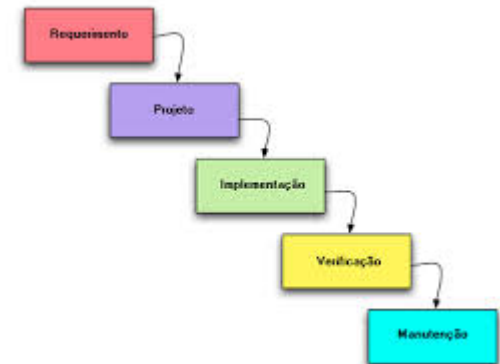
Modelo Cascata





Modelo Cascata

- Requisitos de um problema são bem conhecidos.
- Abordagem sequencial e sistemática para o desenvolvimento do software.
- Inicia-se com o levantamento de necessidades, avançando pelas fases de planejamento, modelagem, construção, implantação e suporte contínuo ao software concluído.
- Paradigma mais antigo da Engenharia de Software.





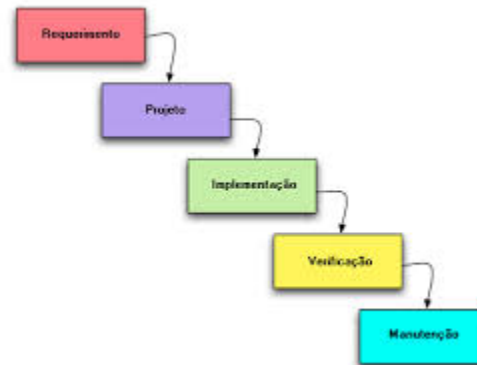
Porque esse modelo é chamado de Modelo Cascata ?





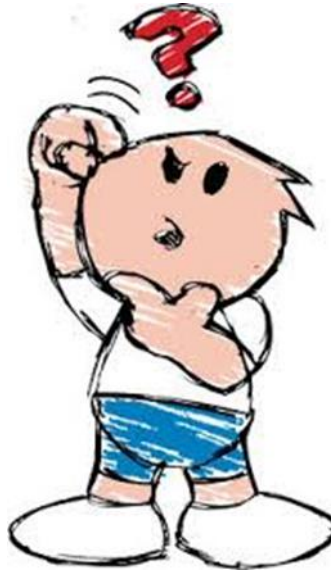
Modelo de Processo Cascata

- As transições de uma atividade para outra se assemelham a uma cascata.
- Esse modelo também é chamado “ciclo de vida clássico de software”





Quais os principais estágios ou fases do modelo cascata ?





Modelo Cascata – Estágios ou Fases

- Definição e Análise de Requisitos
- Projeto do Software
- Implementação e Testes de Unidade
- Integração e Testes de Sistema
- Operação e Manutenção





Quais os principais desafios do Modelo Cascata ?





Modelo Cascata – Desafios

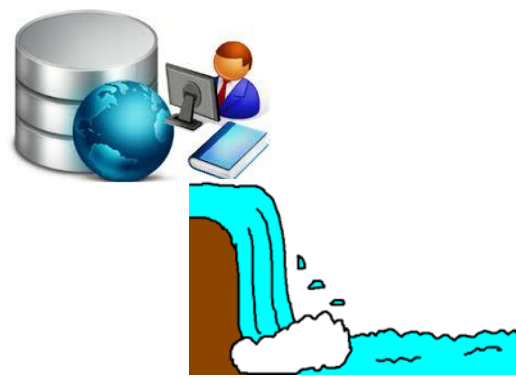
- O principal desafio do modelo cascata é a **dificuldade de acomodar mudanças** após o processo estar em execução.
- Em princípio, uma fase tem de ser concluída antes de se mover para a próxima fase.
- Maior dificuldade em responder à mudança de requisitos.
- Cliente deve ter paciência. Uma versão operacional do software não estará disponível antes de se estar próximo ao final do projeto.





Onde o modelo cascata é mais empregado ?





Modelo Cascata – Aplicações

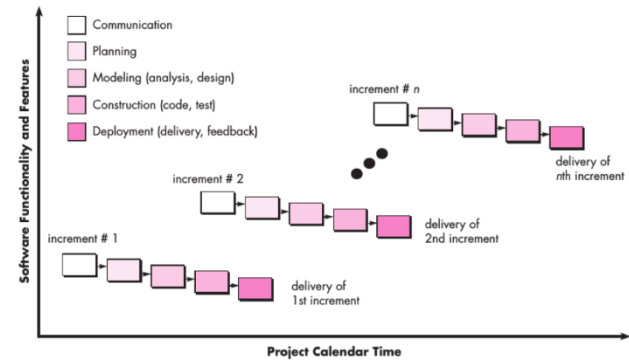
- O modelo é apropriado em sistemas nos quais os requisitos são bem compreendidos e alterações do escopo do sistema estarão limitadas durante a fase de design.
- Poucos sistemas de negócios têm requisitos estáveis.





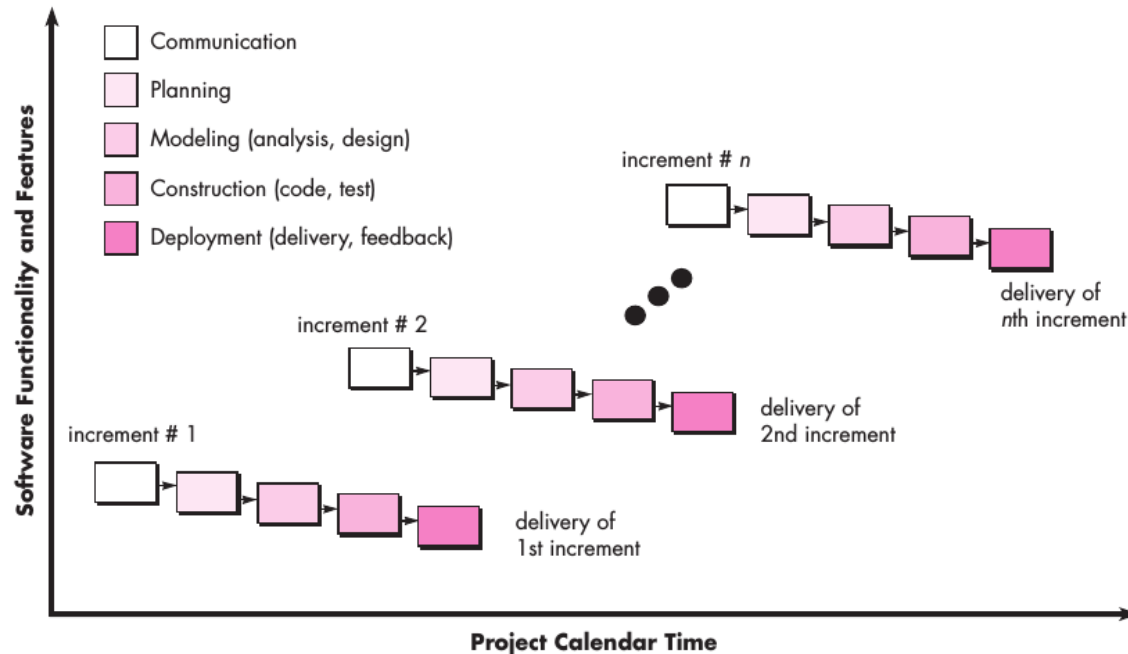
Modelo incremental

- Em várias situações, os requisitos iniciais do software são razoavelmente bem definidos.
- Ao invés de liberar o sistema em uma única entrega, particiona-se o desenvolvimento e a entrega é feita em incrementos.
- Cada incremento conta com uma funcionalidade do sistema.
- Requisitos do usuário são priorizados. Incrementos iniciais contêm implementações dos requisitos de prioridade mais alta.





O modelo incremental

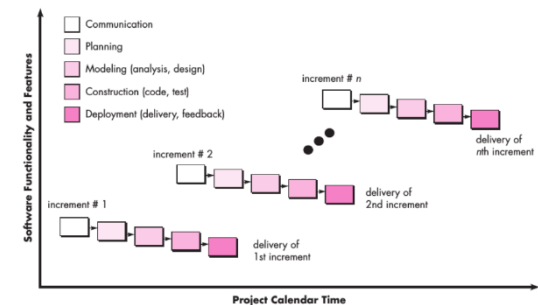


- O modelo incremental combina elementos do fluxo de processos lineares e paralelos.
- Aplicam-se sequências lineares, de forma escalonada, à medida em que o tempo vai avançando.
- Cada sequência linear gera “incrementais” (entregáveis/aprovados/liberados)



Modelo incremental

- Frequentemente, o incremento inicial é um produto essencial, onde os requisitos básicos são atendidos, porém, muitos recursos complementares ainda não são entregues.
- Esse produto essencial é utilizado pelo cliente que pode fazer uma avaliação detalhada.
- Como resultado da avaliação, desenvolve-se um planejamento para implementar-se o próximo incremento.
- Foco do modelo é voltado para a entrega de um produto operacional em cada incremento.
- Útil no planejamento das equipes de trabalho. (equipes podem evoluir com a aprovação dos incrementos...)





Modelo Evolucionário

- Software evolui com o tempo.
- Conforme o desenvolvimento do projeto avança, as necessidades de negócio e de produto podem mudar, tornando inadequado o fluxo linear de atividades.
- Nessas situações, deve-se usar um modelo de processo que tenha sido projetado para desenvolver software que evolua com o tempo.
- Modelos evolucionários são iterativos.



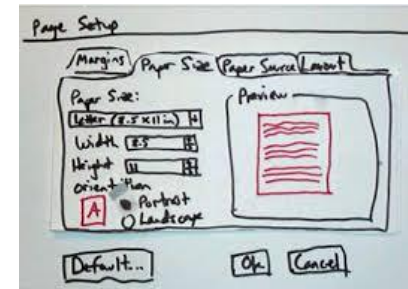
Como desenvolver um software que evolui ao longo do tempo ?





Modelo Evolucionário

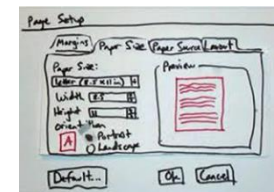
- Prototipação
- Modelo Espiral





Prototipação

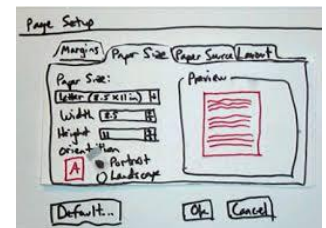
- Cliente pode definir apenas uma série de objetivos gerais para o software.
- Cliente pode não identificar – detalhadamente – os requisitos para funções e recursos do software.
- Em outras situações, o desenvolvedor pode não estar suficientemente seguro quanto à diversas questões relacionadas ao desenvolvimento do software.
- Nessas situações, um protótipo é a melhor abordagem.





Prototipação

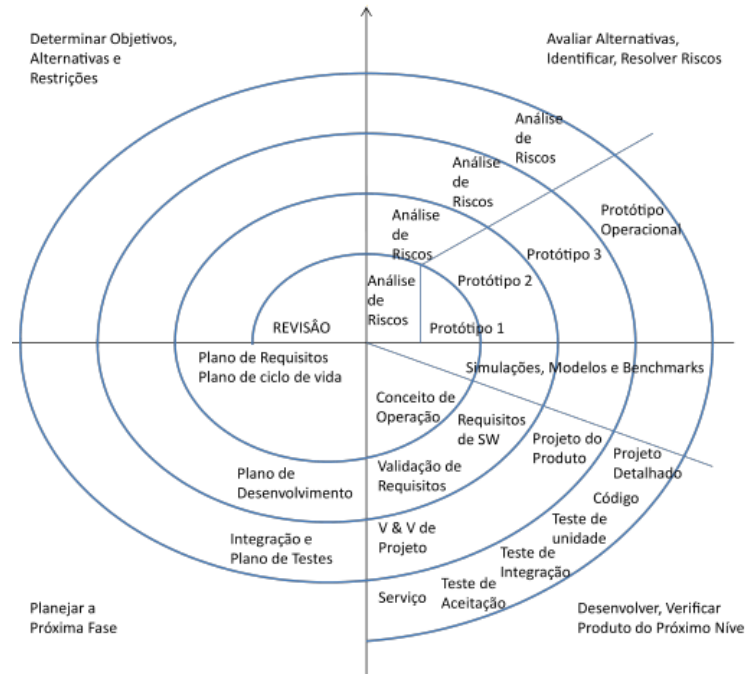
- Auxilia os interessados na compreensão do que será construído.
- A partir dos aspectos visíveis do software, define-se um “projeto rápido”, com layout da interface com o usuário, formatos de tela, relatórios, etc.
- O protótipo é avaliado pelos envolvidos que – a partir de feedbacks – permite que se aprimore (refinamento) dos requisitos. (Requisitos obscuros podem ser elicitados)
- Protótipo pode servir como “primeiro sistema”, podendo ser também descartáveis.





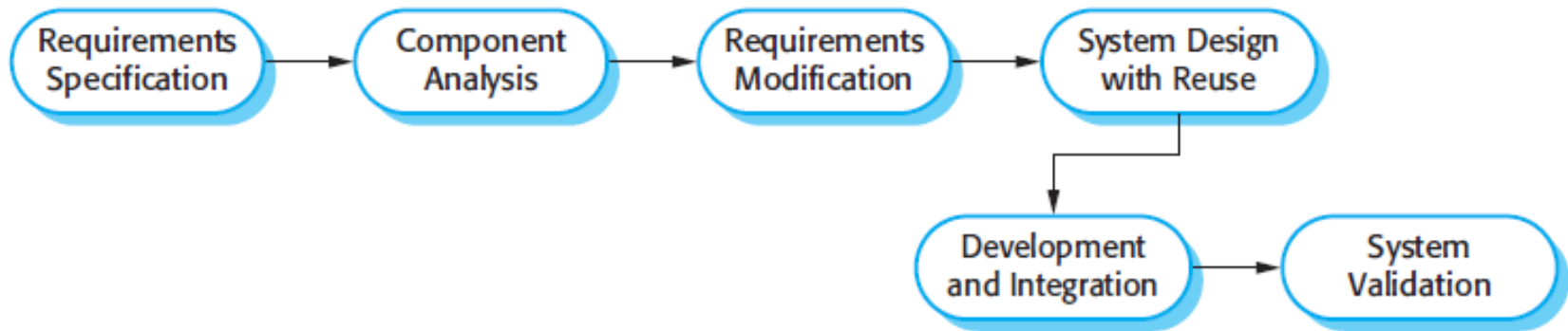
Modelo Espiral

- Modelo de processo evolucionário que acopla a natureza iterativa da prototipação com os aspectos sistemáticos e controlado do modelo cascata.
- O software é desenvolvido em uma série de versões evolucionárias.





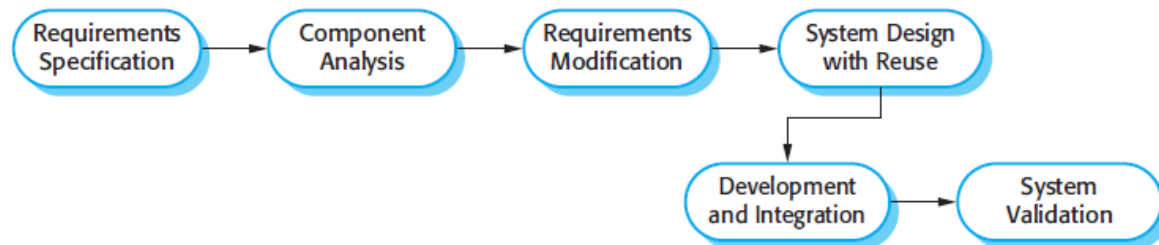
Engenharia de Software orientada a Reuso





Engenharia de Software orientada a Reuso

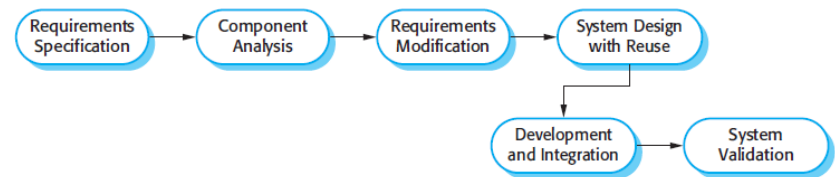
- Na maioria dos projetos de software, há algum reuso de software.
- Isto ocorre de maneira informal, onde pessoas têm conhecimento de designs ou código similar ao que estão necessitando.
- Recentemente, o processo de desenvolvimento de software focado no reuso de software existente tem sido largamente usado.
- Esse modelo baseia-se numa larga base de componentes de software reusável e em um framework de integração para a composição desses componentes.





Estágios do Processo

- **Análise de Componentes** – A partir da especificação, pesquisa-se por componentes que possam implementá-la.
- **Modificação de Requisitos** – Modifica-se os requisitos à luz dos componentes descobertos. Caso seja impossível essa modificação, volta-se para o estágio anterior e pesquisa-se soluções alternativas.
- **Design do Sistema com reuso** – Reusa-se um framework existente ou desenha-se um framework. Se componentes reusáveis não estiverem disponíveis, novo software pode ser desenhado.
- **Desenvolvimento e Integração**





Exemplos – Componentes de Software

- **Web Services** que são desenvolvidos de acordo com padrões de serviços e são disponíveis para invocação remota.
- **Collections** de objetos que são desenvolvidos como packages para serem integrados com um framework de componentes tais como .NET ou JEE.

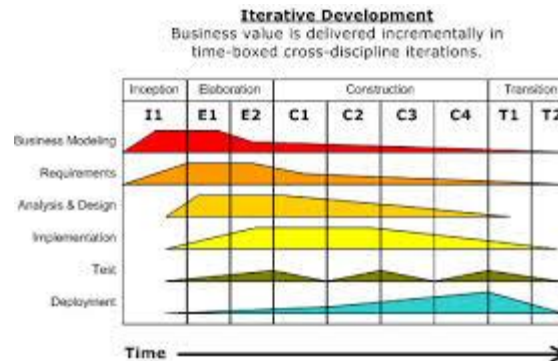




Rational Unified Process



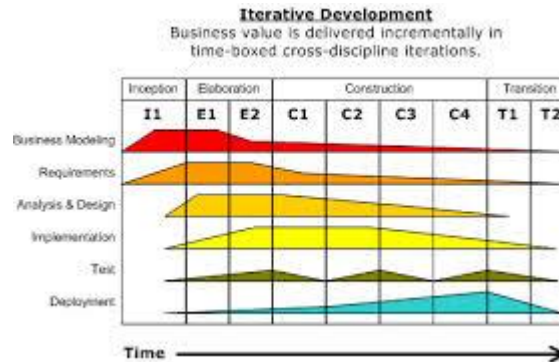
- Um moderno processo genérico derivado do trabalho de UML.
- Normalmente descrito a partir de 3 perspectivas:
 - ✓ Uma perspectiva dinâmica que mostra fases sobre o tempo;
 - ✓ Uma perspectiva estática que mostra atividades do processo;
 - ✓ Uma perspectiva prática a qual sugere boas práticas para serem usadas durante o processo.





Fases do RUP

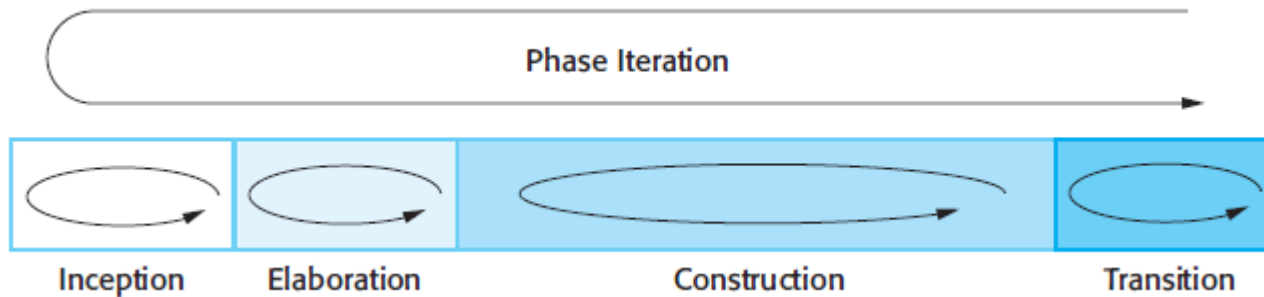
- **Incepção.** Identificam-se as necessidades de negócio para o software. Levantam-se requisitos de negócio.
- **Elaboração.** Desenvolve-se a modelagem do software. Define-se uma solução para o problema do ponto de vista de arquitetura.
- **Construção.** Projeto, programação e testes.
- **Transição.** Libera o sistema para seu ambiente operacional.





RUP – Iteração

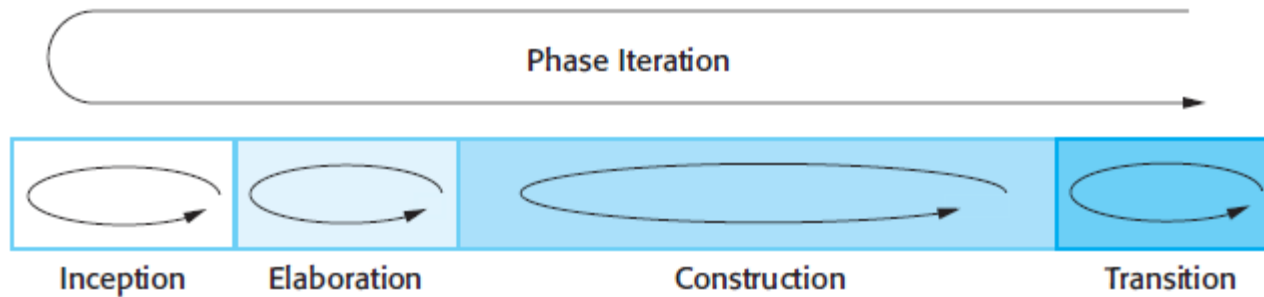
- Cada fase é iterativa com resultados desenvolvidos de forma incremental.
- Iteração entre fases. Todo o conjunto de fases pode ser tratado de forma incremental.





RUP – Boas práticas

- Software desenvolvido de forma iterativa.
- Requisitos devem ser gerenciados.
- Emprego de arquiteturas baseadas em componentes.





RUP – Boas práticas

- Uso de modelos gráficos UML para apresentar visões estáticas e dinâmicas do software.
- Verificação da qualidade de software.
- Gerenciamento das mudanças de software.

