
Processo de Software

Processo de Software

- Conjunto de atividade coerentes para especificar, projetar, implementar e testar sistemas de software

Objetivos

- Introduzir modelo de processo de software
- Descrever diversos modelos de processos e onde eles podem ser usados
- Descrever saídas de modelos de processo para engenharia de requisitos, desenvolvimento, teste e evolução de software
- Introduzir tecnologias CASE para suportar atividades de processo de software

Tópicos Abordados

- Modelos de processo de software
- Iteração de processos
- Especificação de software
- Projeto e implementação de software
- Validação de software
- Evolução de software
- Suporte de processos automatizados

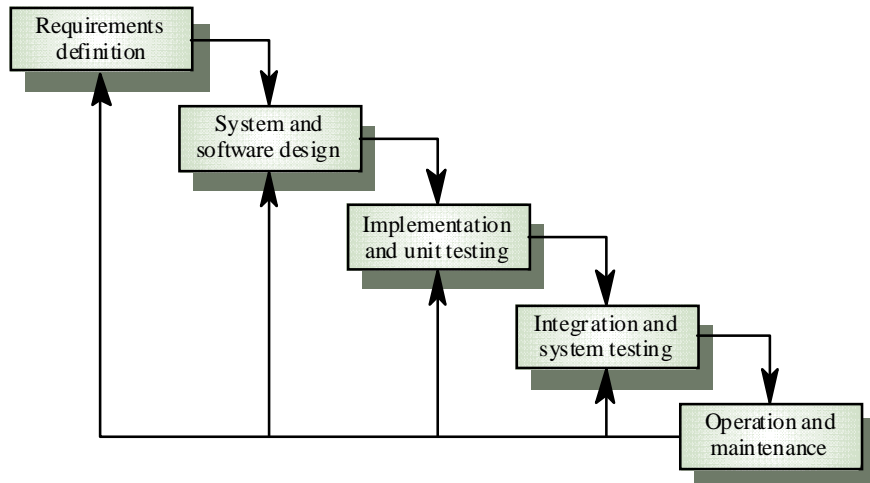
O Processo de Software

- Um conjunto estruturado de atividade requeridas para desenvolver um sistema de software
 - Especificação
 - Projeto
 - Validação
 - Evolução
- Um modelo de processo de software é uma representação abstrata de um processo. Ele representa a descrição de um processo a partir de uma perspectiva qualquer

Modelos de Processo de Software Geral

- Modelo Cascata
 - Fases separadas e distintas de especificação e desenvolvimento
- Desenvolvimento Evolutivo
 - Especificação e desenvolvimento são intercalados
- Desenvolvimento Formal de Sistemas
 - Um modelo matemático de sistema é formalmente transformado em uma implementação
- Desenvolvimento Baseado em Re-uso
 - O sistema é montado a partir de componentes existentes

Modelo em Cascata



Fases do Modelo em Cascata

- Definição e análise de requisitos
- Projeto do sistema e do software
- Implementação e teste dos sub-sistemas
- Integração e teste do sistema como um todo
- Operação e manutenção
- O inconveniente do modelo em cascata é a dificuldade de acomodar as mudanças após o processo estar em andamento

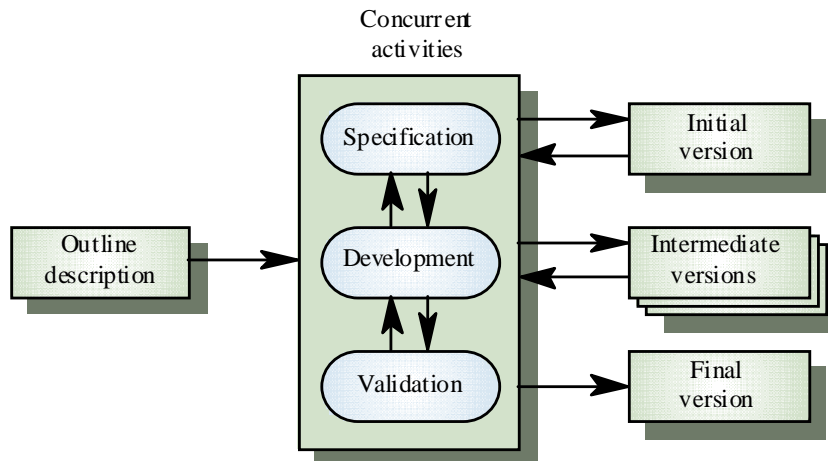
Problemas do Modelo em Cascata

- Particionamento inflexível do projeto em estágios distintos
- Isto faz ele responder com dificuldades a mudanças solicitadas pelo cliente
- Conseqüentemente, este modelo somente é apropriado onde os requisitos são bem compreendidos

Desenvolvimento Evolucionário

- Desenvolvimento exploratório
 - O objetivo é trabalhar com o cliente para evoluir para o sistema final através de um esboço inicial. Se iniciar com requisitos bem compreendidos
- Prototipação descartável
 - O objetivo é entender os requisitos do sistema. Se iniciar com mal entendimento dos requisitos

Desenvolvimento Evolucionário



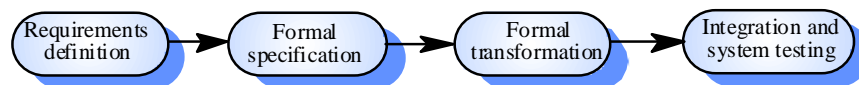
Desenvolvimento Evolucionário

- Problemas
 - Falta de visibilidade do processo
 - O sistema freqüentemente é mal estruturado
 - Conhecimentos especiais (ex.: em linguagens de prototipação rápida) podem ser requeridas
- Aplicabilidade
 - Para sistemas interativos de tamanho pequeno ou médio
 - Para partes de grandes sistemas
 - » Ex.: a interface do usuário
 - Para sistema de curta duração

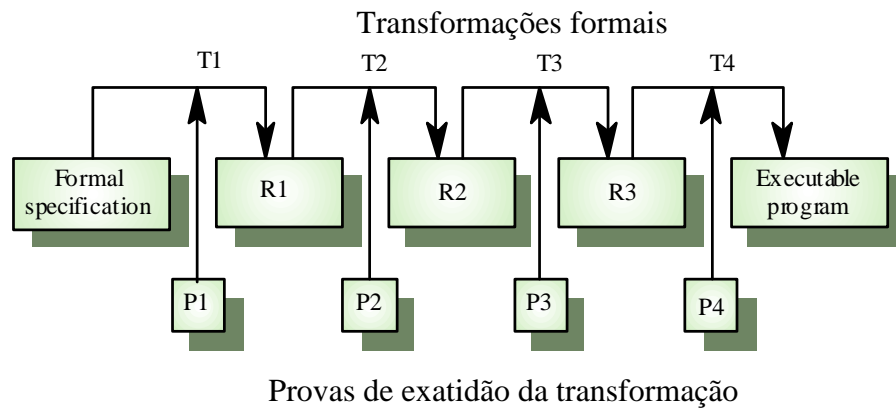
Desenvolvimento de Sistemas Formais

- A especificação de requisitos de software é redefinida em uma especificação formal detalhada, que é expressa em uma notação matemática
- Os processos de desenvolvimento de projeto, implementação e teste de unidade são substituídos por um processo de desenvolvimento transformacional, em que a especificação formal é refinada por meio de uma série de transformações, em um programa

Desenvolvimento de Sistemas Formais



Transformações Formais



Desenvolvimento de Sistemas Formais

- Problemas
 - Precisa de especialistas com conhecimento e treinamento para aplicar as técnicas
 - Dificuldade de especificar formalmente alguns aspectos do sistema, tal como a interface de usuário
 - Aplicabilidade

Desenvolvimento Orientado ao Re-Uso

- Baseado em uma sistemática de re-uso onde os sistemas são integrados a partir de componentes existentes ou sistemas COTS (Commercial-off-the-shelf)
- Embora o estágio inicial de especificação de requisitos e o estágio de validação sejam comparáveis com outros processos, os estágios intermediários em um processo orientado a reuso são diferentes
- Estágios do Processo
 - Análise dos componentes
 - Modificação dos requisitos
 - Projeto do sistema com re-uso
 - Desenvolvimento e integração

Análise de Componentes

- Considerando a especificação de requisitos, é feita uma busca de componentes para implementar essa especificação. Em geral, não existe combinação exata e os componentes que podem ser utilizados fornecem somente parte da funcionalidade requerida

Modificação de Requisitos

- Durante este estágio, os requisitos são analisados, utilizando-se as informações sobre os componentes que foram encontrados. Eles são então modificados para refletir os componentes disponíveis. Quando as modificações forem impossíveis, a atividade de análise de componentes poderá ser refeita, a fim de procurar soluções alternativas

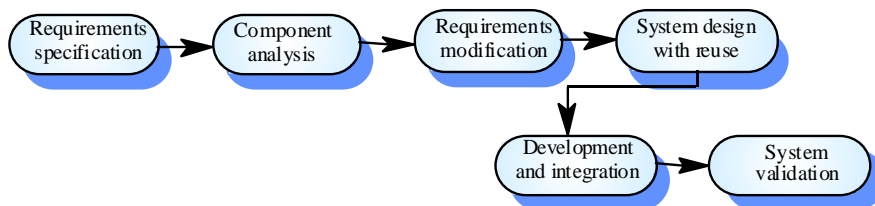
Projeto de Sistema com Reuso

- Durante essa fase, a infra-estrutura do sistema é projetada ou uma infra-estrutura existente é reutilizada. Os projetistas levam em conta os componentes que são reutilizados e organizam a infra-estrutura para lidar com esse aspecto. Um novo software poderá ter de ser projetado, se os componentes reutilizáveis não estiverem disponíveis

Desenvolvimento e Integração

- O software que não puder ser comprado será desenvolvido, e os componentes e sistemas COTS serão integrados, a fim de criar um sistema. A integração de sistemas, nesse modelo, pode ser parte do processo de desenvolvimento, em vez de uma atividade separada

Desenvolvimento Orientado ao Re-Uso



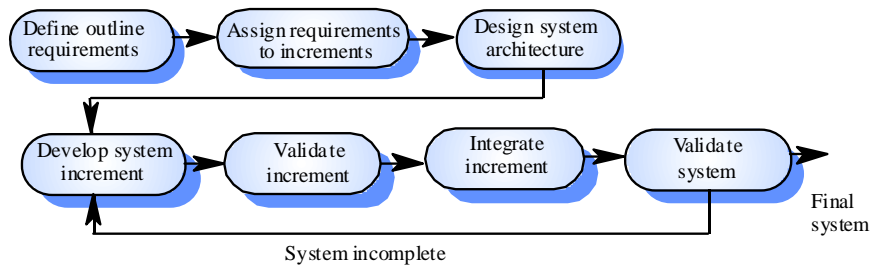
Iteração de Processos

- Requisitos de sistema SEMPRE evoluem durante o curso de um projeto, assim iteração de processos onde os estágios mais adiantados são re-utilizados é sempre parte do processo de um grande sistema
- Iteração pode ser aplicada a qualquer modelo de processo genérico
- Dois modelos híbridos:
 - Desenvolvimento incremental
 - Desenvolvimento espiral

Desenvolvimento Incremental

- Requisitos de usuários são priorizados e, requisitos de alta prioridade são incluídos em incrementos mais adiantados
- Uma vez que o desenvolvimento de um incremento é iniciado, os requisitos são congelados, embora os requisitos para um incremento mais atrasado continuem a evoluir

Desenvolvimento Incremental



Vantagens do Desenvolvimento Incremental

- Os clientes não precisam esperar até que todo o sistema seja entregue, para então tirarem proveito dele
- Os clientes podem utilizar os primeiros incrementos como um protótipo e obter uma experiência que forneça os requisitos para estágios posteriores do sistema
- Baixo risco de falha total do projeto
- Como as funções prioritárias são entregues primeiro e incrementos posteriores são integrados a elas, é inevitável que as funções de sistema mais importantes passem pela maior parte dos testes

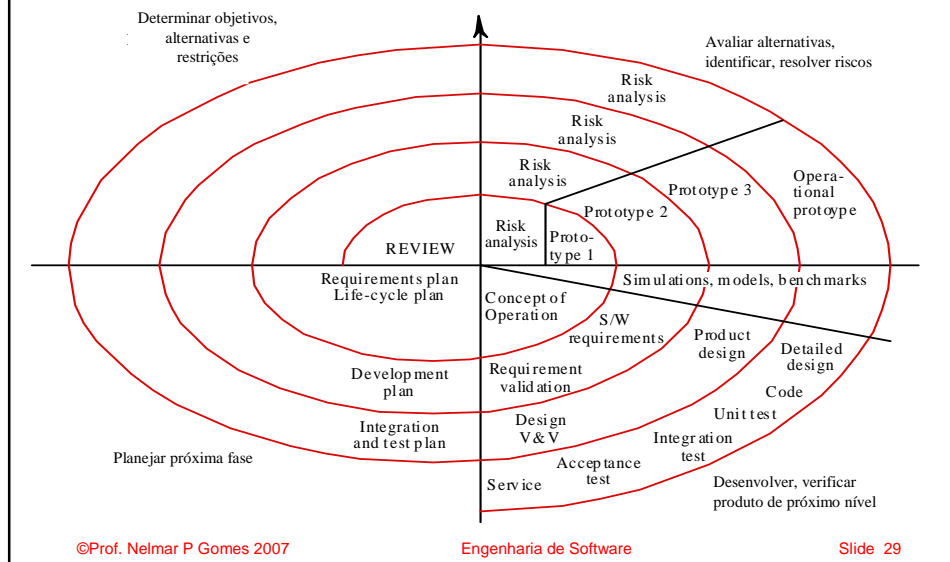
Extreme programming (Programação Extrema)

- Recente evolução dessa abordagem incremental, tem como base o desenvolvimento e a entrega de incrementos de funcionalidades muito pequenos
- Confiança na melhoria constante do código, devido ao envolvimento do cliente no processo, a constante melhoria do código e a programação impessoal

Desenvolvimento Espiral

- O processo é representado como uma espiral em vez de representar o processo como um seqüência de atividades com algum retorno de uma atividade para outra
- Cada volta na espiral representa uma fase no processo
- Fases não fixas tais como, especificação ou projeto – as voltas na espiral são escolhidas dependendo do que é requerido
- Os riscos são avaliados explicitamente e resolvidos durante todo o processo

Modelo Espiral do Processo de Software



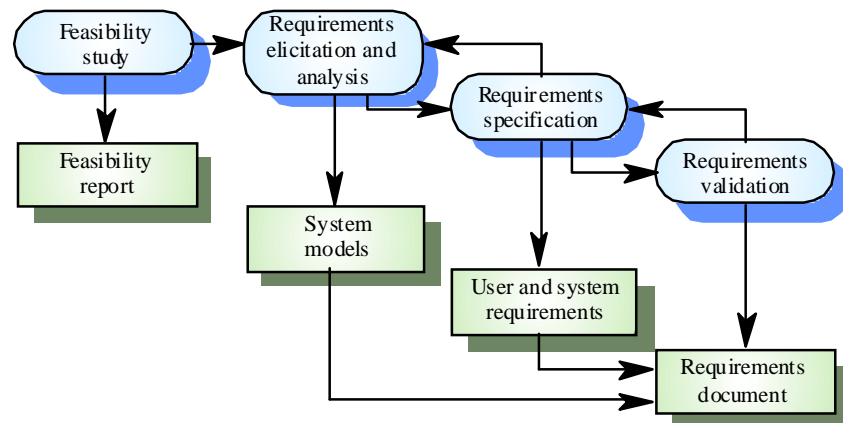
Setores do Modelo Espiral

- **Definição de Objetivos**
 - São definidos os objetivos específicos para essa fase do projeto
- **Avaliação e redução de riscos**
 - Para os riscos identificados, é realizada uma análise detalhada e são tomadas providências para reduzir esse risco
- **Validação e desenvolvimento**
 - Um modelo de desenvolvimento para o sistema é escolhido, o qual pode ser algum dos modelos genéricos
- **Planejamento**
 - O projeto é revisto e a próxima fase do espiral é planejada

Especificação de Software

- O processo de engenharia de requisitos leva à produção de uma documentação de requisitos, que é a especificação do sistema
- Processo de engenharia de requisitos
 - Estudo de viabilidade
 - Levantamento e análise de requisitos
 - Especificação de requisitos
 - Validação de requisitos

O Processo de Engenharia de Requisitos



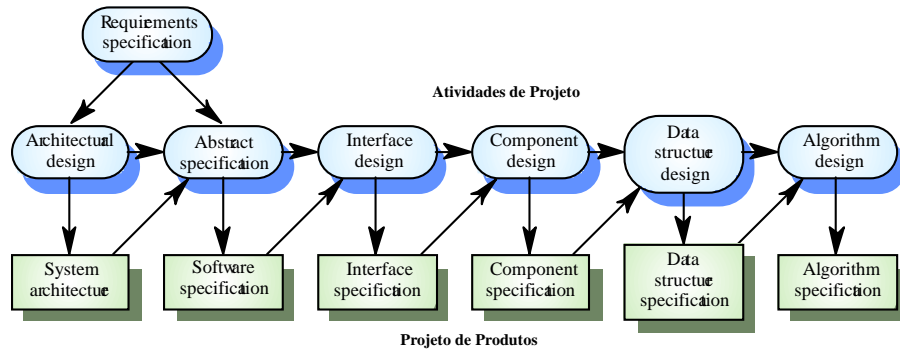
Projeto e Implementação de Software

- O processo de converter a especificação do sistema em um sistema executável
- Projeto de Software
 - Descrição da estrutura do software a ser implementado
- Implementação
 - Traduz essa estrutura em um programa executável
- Uma especificação para o próximo estágio é a saída de cada atividade de projeto. Essa especificação pode ser uma especificação abstrata, formal, produzida para esclarecer os requisitos, ou pode ser uma especificação de como parte do sistema é implementado

Atividades de Processo de Projeto

- Projeto de arquitetura
- Especificação abstrata
- Projeto de interface
- Projeto de componentes
- Projeto de estrutura de dados
- Projeto de algoritmo

The software design process



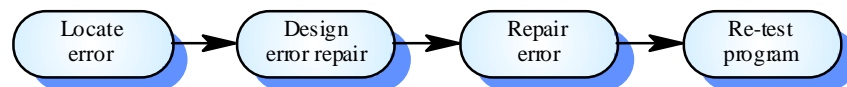
Métodos de Projeto

- Em muitos projetos de desenvolvimento de software, o projeto de software ainda é um processo utilizado à medida que for necessário
- O projeto é geralmente documentado como um conjunto de modelos gráficos
- Modelos possíveis:
 - Modelo de fluxo de dados
 - Modelo de relacionamento de entidades
 - Modelo estrutural
 - Métodos orientados a objeto

Programação e Depuração

- Traduzir um projeto em um programa e remover os erros a partir deste programa
- Programação é uma atividade pessoal, não existe um processo genérico de programação
- Programadores desenvolvem alguns testes do código que eles mesmos desenvolveram, o que revela defeitos que devem ser removidos, esse processo é chamado de depuração

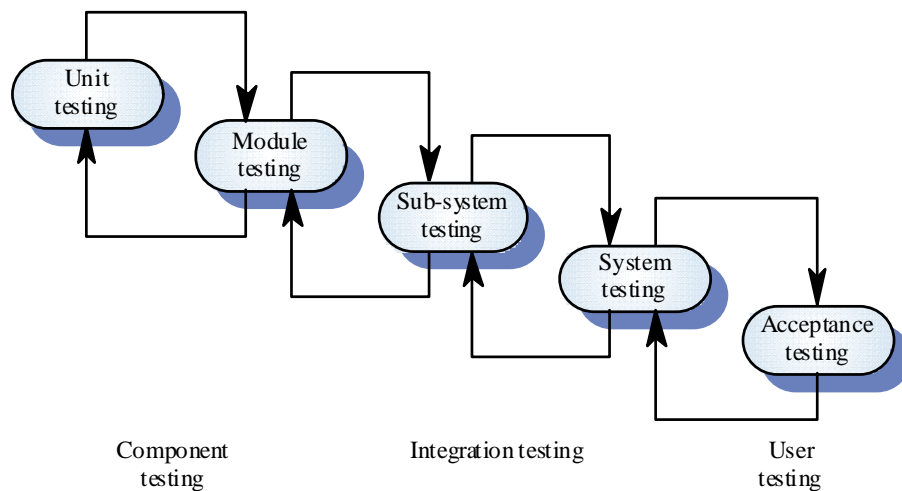
O Processo de Depuração



Validação de Software

- Verificação e Validação (V & V), destina-se a mostrar que um sistema está de acordo com suas especificações e que ele atende às expectativas do cliente comprador do sistema
- Envolve checar e revisar processos e testar o sistema
- Testar o sistema envolve executar o sistema com casos de teste que são derivados da especificação de dados reais a serem processados pelo sistema

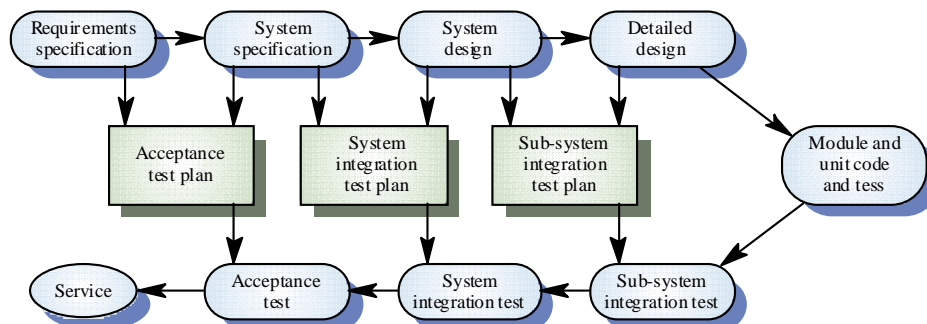
O Processo de Teste



Estágios do Processo de Teste

- Teste de unidade
 - Testa os componentes individuais, para garantir que eles operam corretamente
- Teste de módulo
 - Coleção de componentes dependentes que são testados
- Teste de subsistema
 - Testa conjuntos de módulos que foram integrados. O foco é na detecção de erros de interfaces de módulos, mediante utilização rigorosa destas interfaces
- Teste de sistema
 - Teste do sistema como um todo. Teste das propriedades emergentes
- Teste de Aceitação
 - Teste com dados do cliente para checar se o sistema está aceitável operacionalmente

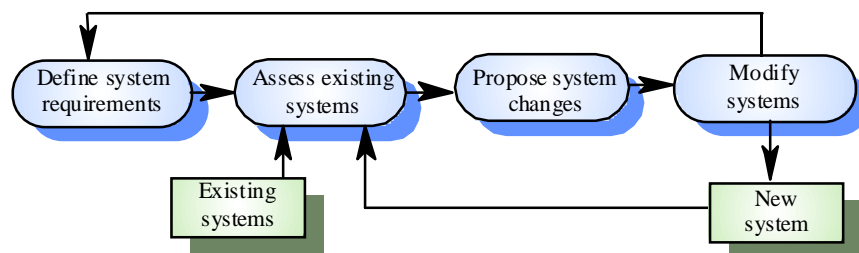
Fases de Teste



Evolução de Software

- O software é flexível e pode mudar
- Os requisitos podem mudar de acordo com as circunstâncias das mudanças do negócio e o software que suporta esse negócio também evolui e muda
- Entre o desenvolvimento e a evolução existe a manutenção. Os processo de manutenção são considerados menos desafiadores do o desenvolvimento de software original

Evolução de Sistema



Apoio ao Processo Automatizado (CASE)

- *Computer-aided software engineering* (CASE) é um software de suporte ao processo de desenvolvimento e evolução do software
- Atividades que podem ser automatizadas:
 - Desenvolvimento de modelos gráficos de sistemas
 - Compreensão de um projeto utilizando-se dicionário de dados que contém informações sobre as entidades e sua relação
 - Geração de interfaces com usuários, a partir de uma descrição gráfica da interface
 - Depuração de programas, pelo fornecimento de informações sobre um programa em execução
 - Tradução automatizada de programas, de uma antiga versão de linguagem de programação para uma mais recente

Tecnologia CASE

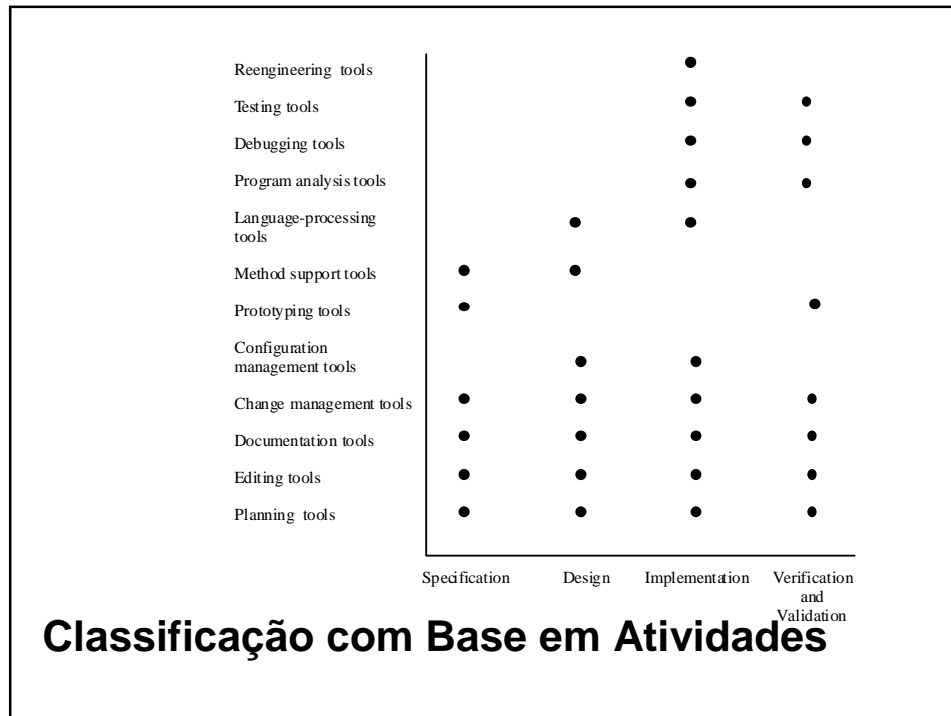
- A tecnologia CASE está atualmente disponível para a maioria das atividades de rotina no processo de software
 - Engenharia de software requer atividades criativas, isto não é possível automatizar
 - A engenharia de software é realizada em equipe, e os engenheiros de software dedicam bastante tempo interagindo com os outros membros da equipe. A tecnologia CASE não é muito compatível com essa situação

Classificação CASE

- A classificação ajuda-nos a entender os diferentes tipos de ferramentas CASE e seu papel em apoiar as atividade de processo de software
- Perspectiva funcional
 - As ferramentas são classificadas de acordo com sua função específica
- Perspectiva de processo
 - As ferramentas são classificadas de acordo com a atividade de processo que elas apóiam
- Perspectiva de integração
 - As ferramentas são classificadas de acordo com a maneira como são organizadas em unidades integradas

Classificação Funcional

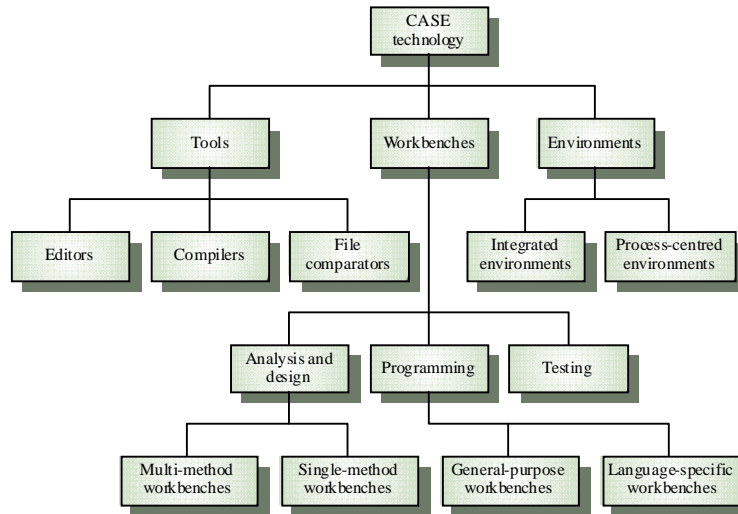
Tipo de Ferramenta	Exemplos
Ferramenta de planejamento	Ferramentas PERT, planilhas eletrônicas
Ferramenta de edição	Editores de texto, editores de diagrama
Ferramenta de gerenciamento de mudança	Ferramentas de controle de requisitos
Ferramenta de gerenciamento de configuração	Sistemas de gerenciamento de versão
Ferramenta de prototipação	Linguagens de nível muito alto, geradores de interface
Ferramenta de apoio a métodos	Editores de projeto, dicionários de dados
Ferramenta de processamento de linguagem	Compiladores, interpretadores
Ferramenta de análise de programa	Geradores de referência cruzada, analisadores estáticos
Ferramenta de testes	Geradores de dados de testes, comparadores de arquivos
Ferramenta de depuração	Sistemas interativos de depuração
Ferramenta de documentação	Programas de layout de página
Ferramenta de reengenharia	Sistemas de reestruturação de programas



Outra Forma de Classificação CASE

- Ferramentas
 - Apóiam as tarefas de processos individuais, como a verificação da consistência de um projeto, a compilação de um programa, entre outras
- *Workbenches*
 - Apóiam as fases ou atividades de processo, como a especificação e o projeto
- Ambientes
 - Oferecem apoio a todo ou, pelo menos, a uma parte substancial do processo de software. Normalmente, incluem vários *workbenches* diferentes, que são integrados de alguma maneira

Ferramentas, *Workbenches*, Ambientes



Pontos-chave

- Os processos de software são atividades envolvidas na produção de um sistema de software. Os modelos de processo de software são representações abstratas desses processos
- Todos os processos de software incluem especificação de software, projeto e implementação, validação e evolução
- Os modelos de processo iterativos apresentam o processo de software com um ciclo de atividades
- A engenharia de requisitos é o processo de desenvolvimento de uma especificação de software

Pontos-chave

- Os processos de projeto e implementação se ocupam de transformação de uma especificação de requisitos em um sistema executável
- A validação de software é o processo de verificar se o sistema está em conformidade com sua especificação e se ele atende às reais necessidades dos usuários do sistema
- A evolução de software se ocupa de modificar os sistemas de software existentes, para que eles atendam a novos requisitos
- A tecnologia CASE proporciona apoio automatizado aos processos de software. As ferramentas CASE oferecem apoio às atividades de processo individuais; as *workbenches* CASE apóiam um conjunto de atividades relacionadas; os ambientes CASE são compatíveis com todas ou com a maioria das atividades de processo de software

Atividades

- Pesquisar e entregar relatório sobre ferramentas CASE
- Responder o questionário da página 58 do livro Engenharia de Software – 6. Edição de Ian Sommerville