Fatec
Araras

Engenharia de Software I

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

"Weeks of coding can save you hours of planning."

Autor desconhecido

Profissão engenheiro de software

Engenheiro de software

Fatec Araras

Assim como outras disciplinas de engenharia, a engenharia de software é desenvolvida dentro de um *framework* social e legal que limita a liberdade das pessoas que trabalham nessa área. Como um engenheiro de software,

Você deve aceitar que seu trabalho envolve maiores responsabilidades do que simplesmente aplicar habilidades técnicas.

Engenheiro de software

Fatec Araras

No entanto, existem áreas nas quais os padrões de comportamento aceitável não são limitados pelas leis, mas pela mais tênue noção de responsabilidade profissional. Algumas delas são:

Confidencialidade. Você deve respeitar naturalmente a confidencialidade de seus empregadores ou clientes, independentemente de ter sido ou não assinado um acordo formal de confidencialidade.

Competência. Você não deve deturpar seu nível de competência. Você não deve aceitar conscientemente um trabalho que esteja fora de sua competência.

Engenheiro de software

Fatec Araras

Direitos de propriedade intelectual. Você deve ter conhecimento das leis locais a respeito da propriedade intelectual, como patentes e *copyright*.

Você deve ter cuidado para garantir que a propriedade intelectual dos empregadores e clientes seja protegida.

Mau uso do computador. Você não deve usar suas habilidades técnicas para fazer mau uso de computadores de outras pessoas. Esse mau uso varia de relativamente trivial (jogar videogames em uma máquina do empregador, por exemplo) até extremamente sério (disseminar vírus ou outros malwares).

Código de ética

Fatec Araras

Quadro 1.1 Código de ética da ACM/IEEE (© IEEE/ACM 1999)

Código de ética e práticas profissionais da engenharia de software

Força-tarefa conjunta da ACM/IEEE-CS para ética e práticas profissionais da engenharia de software

Prefácio

Esta versão reduzida do código resume as aspirações em um alto nível de abstração; as cláusulas que estão inclusas na versão completa fornecem exemplos e detalhes de como essas aspirações mudam a forma como agimos enquanto profissionais de engenharia de software. Sem as aspirações, os detalhes podem se tornar legalistas e tediosos; sem os detalhes, as aspirações podem se tornar altissonantes, porém vazias; juntos, as aspirações e os detalhes formam um código coeso.

Os engenheiros de software devem se comprometer a fazer da análise, especificação, projeto, desenvolvimento, teste e manutenção de software uma profissão benéfica e respeitada. Em conformidade com seu comprometimento com a saúde, a segurança e o bem-estar públicos, engenheiros de software devem aderir a oito princípios:

1. PÚBLICO — Engenheiros de software devem agir de acordo com o interesse público.
2. CLIENTE E EMPREGADOR — Engenheiros de software devem agir de maneira que seja do melhor interesse de seu cliente e empregador e de acordo com o interesse público.
3. PRODUTO — Engenheiros de software devem garantir que seus produtos e modificações relacionadas atendam aos mais altos padrões profissionais possíveis.
4. JULGAMENTO — Engenheiros de software devem manter a integridade e a independência em seu julgamento profissional.
5. GERENCIAMENTO — Gerentes e líderes de engenharia de software devem aceitar e promover uma abordagem ética para o gerenciamento de desenvolvimento e manutenção de software.
6. PROFISSÃO — Engenheiros de software devem aprimorar a integridade e a reputação da profissão de acordo com o interesse público.
7. COLEGAS — Engenheiros de software devem auxiliar e ser justos com seus colegas.
8. SI PRÓPRIO — Engenheiros de software devem participar da aprendizagem contínua durante toda a vida, e devem promover uma abordagem ética para a prática da profissão.

Fonte: Sommerville

Modelos de Processo de software

Processo de software

Fatec Araras

Um processo de software é uma sequência de atividades que leva à produção de um produto de software. Existem quatro atividades fundamentais comuns a todos os processos de software.

1) **Especificação de software**, em que clientes e engenheiros definem o software a ser produzido e as restrições de sua operação.
2) **Desenvolvimento de software**, em que o software é projetado e programado.
3) **Validação de software**, em que o software é verificado para garantir que é o que o cliente quer.
4) **Evolução de software**, em que o software é modificado para refletir a mudança de requisitos do cliente e do mercado.

Processo de software

Fatec Araras

De alguma forma, essas atividades fazem parte de todos os processos de software. Na prática, são atividades complexas em si mesmas, que incluem subatividades como validação de requisitos, projeto de arquitetura, testes unitários etc. Existem também as atividades que dão apoio ao processo, como documentação e gerenciamento de configuração de software.

Os processos de software são complexos e, como todos os processos intelectuais e criativos, dependem de pessoas para tomar decisões e fazer julgamentos. Não existe um processo ideal, a maioria das organizações desenvolve os próprios processos de desenvolvimento de software.

Processo de software

Os processos de software, às vezes, são categorizados como dirigidos a planos ou processos ágeis.

Processos dirigidos a planos são aqueles em que todas as atividades são planejadas com antecedência, e o progresso é avaliado por comparação com o planejamento inicial.

Modelos de Processo de software

Fatec Araras

Um modelo de processo de software é **uma representação simplificada** de um processo de software.

Modelos genéricos não são descrições definitivas dos processos de software. Pelo contrário, são **abstrações** que podem ser usadas para explicar diferentes abordagens de desenvolvimento de software.

Você pode vê-los como frameworks de processos que podem ser ampliados e adaptados para criar processos de engenharia de software mais específicos.

Os modelos de processo mais comuns

Fatec Araras

O **modelo em cascata** considera as atividades fundamentais do processo de especificação, desenvolvimento, validação e evolução, e representa cada uma delas como fases distintas.

O **desenvolvimento incremental** intercala as atividades de especificação, desenvolvimento e validação. O sistema é desenvolvido como uma série de versões (incrementos), de maneira que cada versão adiciona funcionalidade à anterior.

Já a **Engenharia de software orientada a reúso** é baseada na existência de um número significativo de componentes reusáveis. O processo de desenvolvimento do sistema concentra-se na integração desses componentes em um sistema já existente em vez de desenvolver um sistema a partir do zero.

Os modelos de processo mais comuns

Fatec Araras

Esses modelos **não são mutuamente exclusivos** e muitas vezes são usados em conjunto, especialmente para o desenvolvimento de sistemas de grande porte.

Os modelos de processo mais comuns

Fatec Araras

Para sistemas de grande porte, faz sentido combinar algumas das melhores características do modelo em cascata e dos modelos de desenvolvimento incremental. É preciso ter informações sobre os requisitos essenciais do sistema para projetar uma arquitetura de software que dê suporte a esses requisitos. Você não pode desenvolver isso incrementalmente. Os subsistemas dentro de um sistema maior podem ser desenvolvidos com diferentes abordagens.

Modelo Cascata

Modelo Cascata

Fatec Araras

O primeiro modelo do processo de desenvolvimento de software a ser publicado foi derivado de processos mais gerais da engenharia de sistemas (ROYCE, 1970)

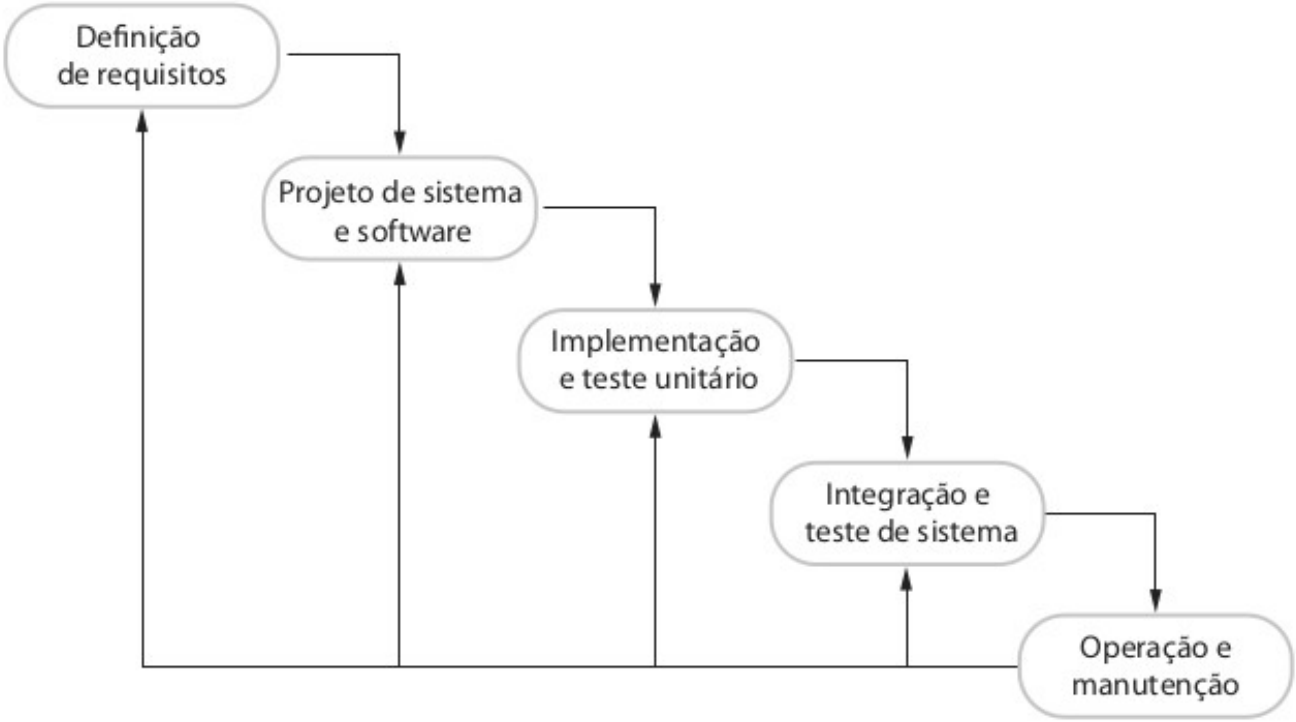
Por causa do encadeamento entre uma fase e outra, esse modelo é conhecido como ‘modelo em cascata’, ou ciclo de vida de software.

O modelo em cascata é um exemplo de um processo dirigido a planos — em princípio, você deve planejar e programar todas as atividades do processo antes de começar a trabalhar nelas.

Modelo Cascata

Fatec Araras

Figura 2.1 O modelo em cascata



Fonte: Sommerville

Modelo Cascata
Principais estágios

Fatec Araras

Os principais estágios do modelo em cascata refletem diretamente as atividades fundamentais do desenvolvimento:

1. Análise e definição de requisitos. Os serviços, restrições e metas do sistema são estabelecidos por meio de consulta aos usuários. Em seguida, são definidos em detalhes e funcionam como uma especificação do sistema.

2. Projeto de sistema e software. O processo de projeto de sistemas aloca os requisitos tanto para sistemas de hardware como para sistemas de software, por meio da definição de uma arquitetura geral do sistema.

Modelo Cascata
Principais estágios

Fatec Araras

3. Implementação e teste unitário. Durante esse estágio, o projeto do software é desenvolvido como um conjunto de programas ou unidades de programa. O teste unitário envolve a verificação de que cada unidade atenda a sua especificação.

4. Integração e teste de sistema. As unidades individuais do programa ou programas são integradas e testadas como um sistema completo para assegurar que os requisitos do software tenham sido atendidos. Após o teste, o sistema de software é entregue ao cliente.

Modelo Cascata
Principais estágios

Fatec Araras

5. Operação e manutenção. Normalmente (embora não necessariamente), essa é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em uso. A manutenção envolve a correção de erros que não foram descobertos em estágios iniciais do ciclo de vida, com melhora da implementação das unidades do sistema e ampliação de seus serviços em resposta às descobertas de novos requisitos.

Modelo Cascata

Em princípio, o resultado de cada estágio é a aprovação de um ou mais documentos (‘assinados’). O estágio seguinte não deve ser iniciado até que a fase anterior seja concluída. Na prática, esses estágios se sobrepõem e alimentam uns aos outros de informações. Durante o projeto, os problemas com os requisitos são identificados durante a codificação, problemas de projeto são encontrados e assim por diante. O processo de software não é um modelo linear simples, mas envolve o feedback de uma fase para outra. Assim, os documentos produzidos em cada fase podem ser modificados para refletirem as alterações feitas em cada um deles.

Modelo Cascata

Fatec Araras

Por causa dos custos de produção e aprovação de documentos, as iterações podem ser dispendiosas e envolver significativo retrabalho.

A solução dos problemas fica para mais tarde, ignorada ou programada, quando possível. Esse congelamento pré-maturo dos requisitos pode significar que o sistema não fará o que o usuário quer. Também pode levar a sistemas mal estruturados, quando os problemas de projeto são contornados por artifícios de implementação.

Modelo Cascata

O modelo em cascata é consistente com outros modelos de processos de engenharia, e a documentação é produzida em cada fase do ciclo.

Em princípio, o modelo em cascata deve ser usado apenas quando os requisitos são bem compreendidos e pouco provavelmente venham a ser radicalmente alterados durante o desenvolvimento do sistema.

Desenvolvimento Incremental

Desenvolvimento Incremental

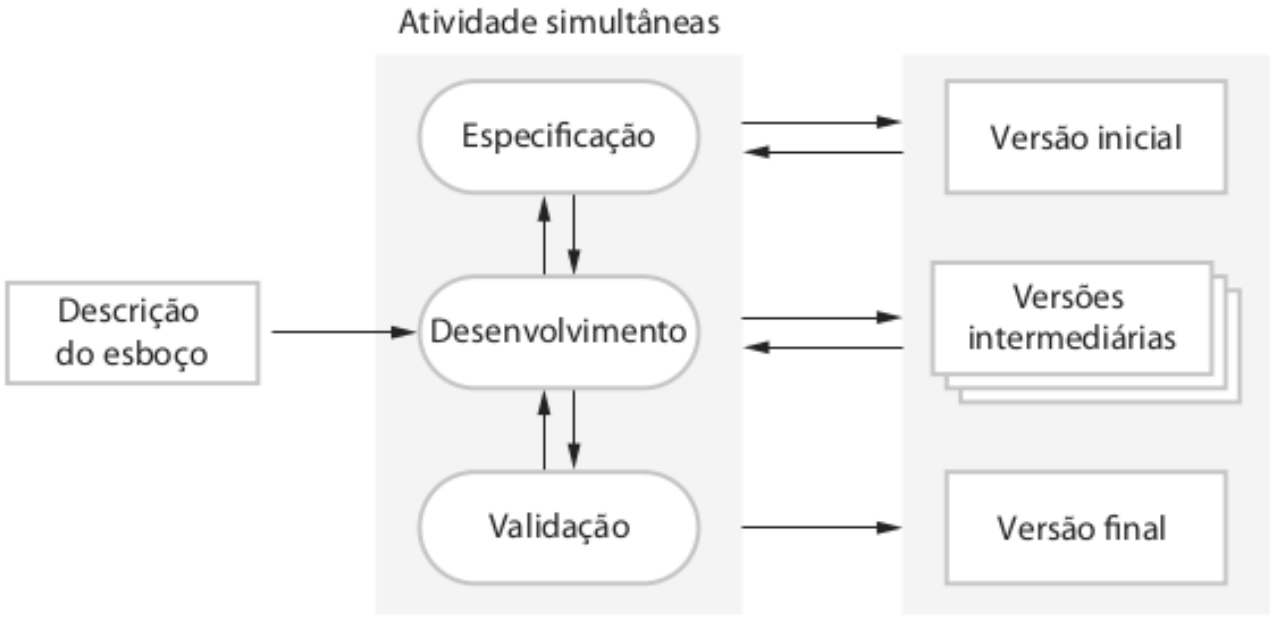
Fatec Araras

O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido.

Atividades de especificação, desenvolvimento e validação são intercaladas, e não separadas, com rápido feedback entre todas as atividades.

Desenvolvimento Incremental

Figura 2.2 Desenvolvimento incremental



Fonte: Sommerville

Desenvolvimento Incremental

Fatec Araras

Desenvolvimento incremental de software, que é uma parte fundamental das abordagens ágeis, é melhor do que uma abordagem em cascata para a maioria dos sistemas de negócios, *e-commerce* e sistemas pessoais.

Desenvolvimento Incremental

Fatec Araras

Desenvolvimento incremental reflete a maneira como resolvemos os problemas.

Raramente elaboramos uma completa solução do problema com antecedência; geralmente movemo-nos passo a passo em direção a uma solução, recuando quando percebemos que cometemos um erro. Ao desenvolver um software de forma incremental, é mais barato e mais fácil fazer mudanças no software durante seu desenvolvimento.

Desenvolvimento Incremental

Fatec Araras

O desenvolvimento incremental tem três vantagens importantes quando comparado ao modelo em cascata:

1. O custo de acomodar as mudanças nos requisitos do cliente é reduzido. A quantidade de análise e documentação a ser refeita é muito menor do que o necessário no modelo em cascata.

2. É mais fácil obter feedback dos clientes sobre o desenvolvimento que foi feito. Os clientes podem fazer comentários sobre as demonstrações do software e ver o quanto foi implementado. Os clientes têm dificuldade em avaliar a evolução por meio de documentos de projeto de software.

3. É possível obter entrega e implementação rápida de um software útil ao cliente, mesmo se toda a funcionalidade não for incluída. Os clientes podem usar e obter ganhos a partir do software inicial antes do que é possível com um processo em cascata

Desenvolvimento Incremental

Fatec Araras

Do ponto de vista do gerenciamento, a abordagem incremental tem dois problemas:

1. O processo não é visível. Os gerentes precisam de entregas regulares para mensurar o progresso. Se os sistemas são desenvolvidos com rapidez, não é economicamente viável produzir documentos que reflitam cada uma das versões do sistema.

2. A estrutura do sistema tende a se degradar com a adição dos novos incrementos. A menos que tempo e dinheiro sejam dispendidos em refatoração para melhoria do software, as constantes mudanças tendem a corromper sua estrutura. Incorporar futuras mudanças do software torna-se cada vez mais difícil e oneroso.

Os problemas do desenvolvimento incremental são particularmente críticos para os sistemas de vida-longa, grandes e complexos, nos quais várias equipes desenvolvem diferentes partes do sistema.

Engenharia de software orientada a reúso

Engenharia de Software orientada a reúso

Fatec Araras

Na maioria dos projetos de software, há algum reúso de software. Isso acontece muitas vezes informalmente, quando as pessoas envolvidas no projeto sabem de projetos ou códigos semelhantes ao que é exigido.

Esse reúso informal ocorre independentemente do processo de desenvolvimento que se use.

Engenharia de Software orientada a reúso

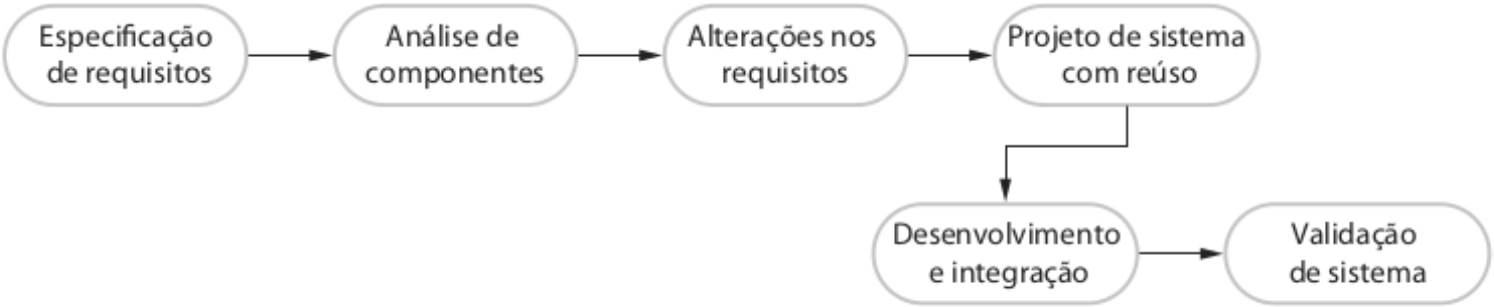
Fatec Araras

No entanto, no século XXI, processos de desenvolvimento de software com foco no reúso de software existente tornaram-se amplamente usados. Abordagens orientadas a reúso dependem de uma ampla base de componentes reusáveis de software e de um framework de integração para a composição desses componentes.

Engenharia de Software orientada a reúso

Fatec Araras

Figura 2.3 Engenharia de software orientada a reúso



Fonte: Sommerville

Engenharia de Software orientada a reúso

Fatec Araras

Embora o estágio de especificação de requisitos iniciais e o estágio de validação sejam comparáveis a outros processos de software, os estágios intermediários em um processo orientado a reúso são diferentes.

Engenharia de Software orientada a reúso

Fatec Araras

1. Análise de componentes. Dada a especificação de requisitos, é feita uma busca por componentes para implementar essa especificação. Em geral, não há correspondência exata, e os componentes que podem ser usados apenas fornecem alguma funcionalidade necessária.

2. Modificação de requisitos. Durante esse estágio, os requisitos são analisados usando-se informações sobre os componentes que foram descobertos. Em seguida, estes serão modificados para refletir os componentes disponíveis. No caso de modificações impossíveis, a atividade de análise dos componentes pode ser reinserida na busca por soluções alternativas.

Engenharia de Software orientada a reúso

Fatec Araras

3. Projeto do sistema com reúso. Durante esse estágio, o framework do sistema é projetado ou algo existente é reusado. Os projetistas têm em mente os componentes que serão reusados e organizam o framework para reúso.

Alguns softwares novos podem ser necessários, se componentes reusáveis não estiverem disponíveis.

4. Desenvolvimento e integração. Softwares que não podem ser adquiridos externamente são desenvolvidos, e os componentes e sistemas COTS são integrados para criar o novo sistema. A integração de sistemas, nesse modelo, pode ser parte do processo de desenvolvimento, em vez de uma atividade separada.

Engenharia de Software orientada a reúso

Fatec Araras

3. Projeto do sistema com reúso. Durante esse estágio, o framework do sistema é projetado ou algo existente é reusado. Os projetistas têm em mente os componentes que serão reusados e organizam o framework para reúso.

Alguns softwares novos podem ser necessários, se componentes reusáveis não estiverem disponíveis.

4. Desenvolvimento e integração. Softwares que não podem ser adquiridos externamente são desenvolvidos, e os componentes e sistemas COTS são integrados para criar o novo sistema. A integração de sistemas, nesse modelo, pode ser parte do processo de desenvolvimento, em vez de uma atividade separada.

Engenharia de Software orientada a reúso

Fatec Araras

Engenharia de software orientada a reúso tem a vantagem óbvia de reduzir a quantidade de software a ser desenvolvido e, assim, reduzir os custos e riscos. Geralmente, também proporciona a entrega mais rápida do software. No entanto, compromisos com os requisitos são inevitáveis, e isso pode levar a um sistema que não atende às reais necessidades dos usuários. Além disso, algum controle sobre a evolução do sistema é perdido, pois as novas versões dos componentes reusáveis não estão sob o controle da organização que os está utilizando.

Atividade do processo

Atividade do processo

Atividades técnicas, de colaboração e de gestão.

Objetivo: especificar, projetar e testar o software.

Diferentes ferramentas podem ser usadas como apoio.

Atividades básicas:

- Especificação
- Desenvolvimento
- Validação
- Evolução

Atividade do processo
Especificação de software

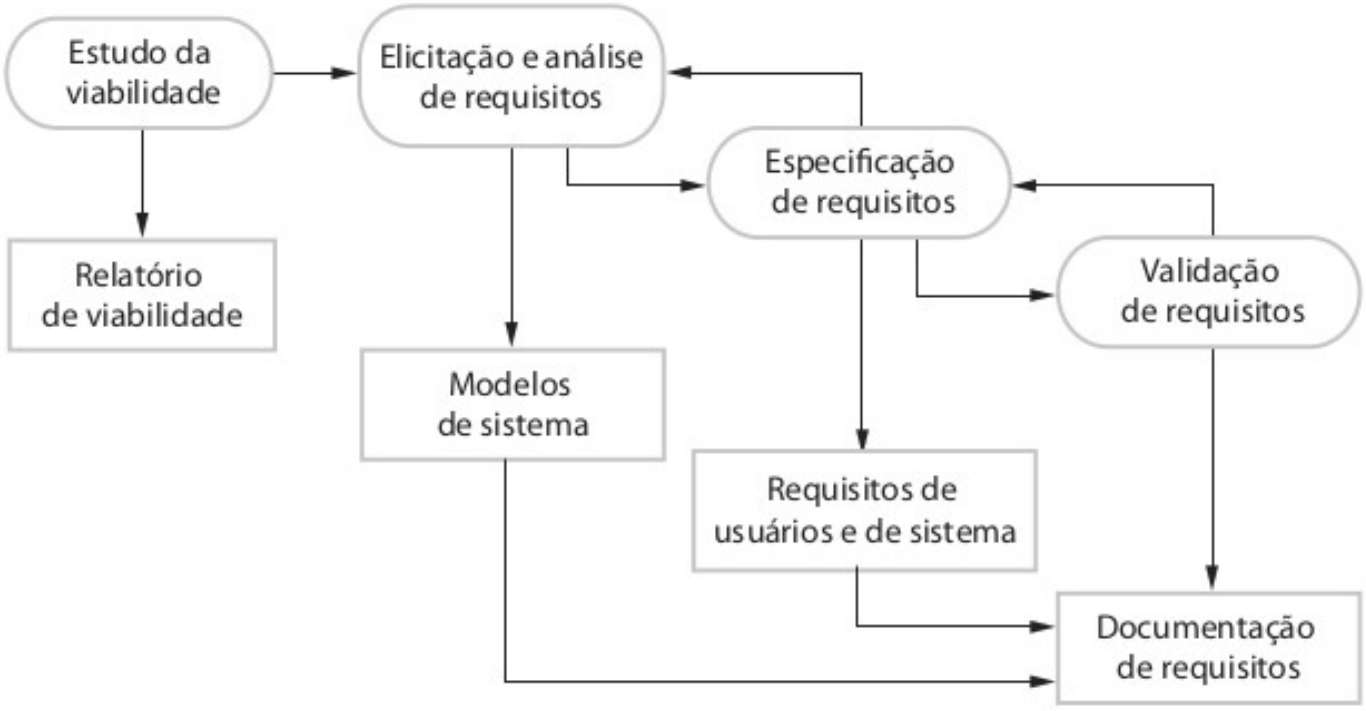
Fatec Araras

Especificação de software ou engenharia de requisitos é o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema. A engenharia de requisitos é um estágio particularmente crítico do processo de software, pois erros nessa fase inevitavelmente geram problemas no projeto e na implementação do sistema.

Atividade do processo
Especificação de software

Fatec Araras

Figura 2.4 Os requisitos da engenharia de processos



Fonte: Sommerville

Atividade do processo
Projeto e implementação

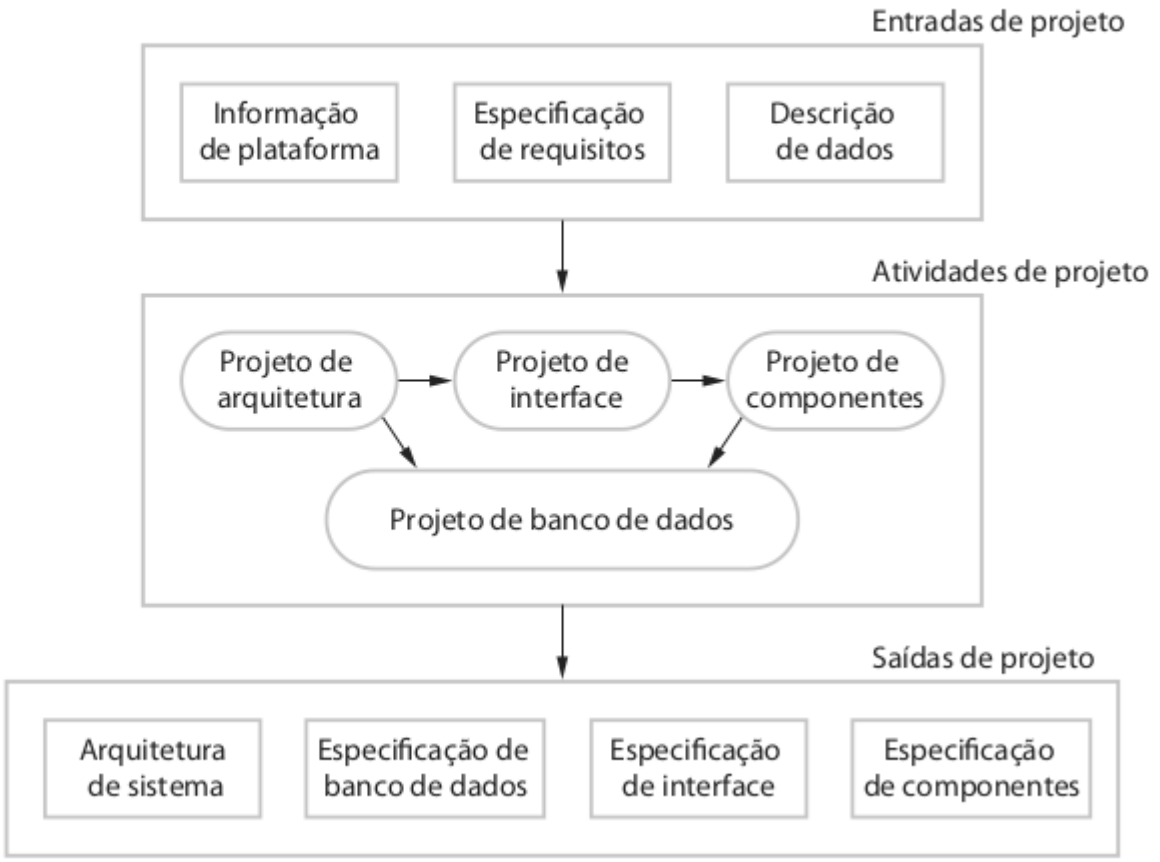
Fatec Araras

O estágio de implementação do desenvolvimento de software é o processo de conversão de uma especificação do sistema em um sistema executável. Sempre envolve processos de projeto e programação de software, mas, se for usada uma abordagem incremental para o desenvolvimento, também pode envolver o refinamento da especificação do software.

Atividade do processo
Projeto e implementação

Fatec Araras

Figura 2.5 Um modelo geral do processo de projeto



Fonte: Sommerville

Atividade do processo
Projeto e implementação

Fatec Araras

As atividades no processo de projeto podem variar, dependendo do tipo de sistema a ser desenvolvido.

1. Projeto de arquitetura, no qual você pode identificar a estrutura geral do sistema, os componentes principais (algumas vezes, chamados subsistemas ou módulos), seus relacionamentos e como eles são distribuídos.

2. Projeto de interface, no qual você define as interfaces entre os componentes do sistema. Essa especificação de interface deve ser inequívoca. Com uma interface precisa, um componente pode ser usado de maneira queoutros componentes não precisam saber como ele é implementado. Uma vez que as especificações de interface são acordadas, os componentes podem ser projetados e desenvolvidos simultaneamente.

Atividade do processo
Projeto e implementação

Fatec Araras

3. Projeto de componente, no qual você toma cada componente do sistema e projeta seu funcionamento. Pode-se tratar de uma simples declaração da funcionalidade que se espera implementar, com o projeto específico para cada programador.

4. Projeto de banco de dados, no qual você projeta as estruturas de dados do sistema e como eles devem ser representados em um banco de dados.

Atividade do processo
Projeto e implementação

Fatec Araras

Essas atividades conduzem a um conjunto de saídas do projeto. Se uma abordagem dirigida a modelos é usada, essas saídas podem ser majoritariamente diagramas. Quando os métodos ágeis de desenvolvimento são usados, as saídas do processo de projeto podem não ser documentos de especificação separado, mas ser representadas no código do programa.

Métodos estruturados para projeto foram desenvolvidos nas décadas de 1970 e 1980, e foram os precursores da UML e do projeto orientado a objetos (BUDGEN, 2003)

Atividade do processo
Projeto e implementação

Fatec Araras

Programação é uma atividade pessoal, não existe um processo geral a ser seguido. Alguns programadores começam com componentes que eles compreendem, desenvolvem-nos e depois passam para os componentes menos compreendidos. Outros preferem a abordagem oposta, deixando para o fim os componentes familiares, pois sabem como desenvolvê-los. Alguns desenvolvedores preferem definir os dados no início do processo e, em seguida, usam essa definição para dirigir o desenvolvimento do programa; outros deixam os dados não especificados durante o maior período de tempo possível.

Atividade do processo
Validação

Fatec Araras

Validação de software ou, mais genericamente, verificação e validação (V&V), tem a intenção de mostrar que um software se adequa a suas especificações ao mesmo tempo que satisfaz as especificações do cliente do sistema.

A validação também pode envolver processos de verificação, como inspeções e revisões, em cada estágio do processo de software, desde a definição dos requisitos de usuários até o desenvolvimento do programa

Atividade do processo
Validação

Fatec Araras

Figura 2.6 Estágios de testes

Fonte: Sommerville

Atividade do processo
Validação

Os estágios do processo de teste são:

1. Testes de desenvolvimento. Os componentes do sistema são testados pelas pessoas que o desenvolveram. Cada componente é testado de forma independente, separado dos outros.

2. Testes de sistema. Componentes do sistema são integrados para criar um sistema completo. Esse processo se preocupa em encontrar os erros resultantes das interações inesperadas entre componentes e problemas de interface do componente.

Atividade do processo
Validação

3. Testes de aceitação. Esse é o estágio final do processo de testes, antes que o sistema seja aceito para uso operacional. O sistema é testado com dados fornecidos pelo cliente, e não com dados advindos de testes simulados.

O teste de aceitação pode revelar erros e omissões na definição dos requisitos do sistema, pois dados reais exercitam o sistema de formas diferentes dos dados de teste.

Fatec
Araras

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Pesquisa

Assistir ao vídeo

Esqueça Metodologias "Ágeis" | [Rated R]

https://www.youtube.com/watch?v=xjjX3R2WuoM