Fatec
Araras

Engenharia de Software I

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

“Se construtores de edifícios construíssem seus prédios como os programadores escrevem seus programas, o primeiro pica-pau que viesse poderia destruir uma civilização”

Gerald Weinberg

Lidando com a mudança

Lidando com mudanças

Fatec Araras

A mudança é inevitável em todos os grandes projetos de software. Os requisitos do sistema mudam, ao mesmo tempo que o negócio que adquiriu o sistema responde a pressões externas e mudam as prioridades de gerenciamento.

A mudança aumenta os custos de desenvolvimento de software, porque geralmente significa que o trabalho deve ser refeito.

Lidando com mudanças

Fatec Araras

Existem duas abordagens que podem ser adotadas para a redução de custos de retrabalho:

1) Prevenção de mudanças, em que o processo de software inclui atividades capazes de antecipar as mudanças possíveis antes que seja necessário qualquer retrabalho.

2) Tolerância a mudanças, em que o processo foi projetado para que as mudanças possam ser acomodadas a um custo relativamente baixo. Isso normalmente envolve alguma forma de desenvolvimento incremental.

Prototipação

Fatec Araras

Um protótipo é uma versão inicial de um sistema de software, usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções. O desenvolvimento rápido e iterativo do protótipo é essencial para que os custos sejam controlados e os *stakeholders* do sistema possam experimentá-lo no início do processo de software.

Prototipação

Fatec Araras

Um protótipo de software pode ser usado em um processo de desenvolvimento de software para ajudar a antecipar as mudanças que podem ser requisitadas:

1) No processo de engenharia de requisitos, um protótipo pode ajudar na elicitação e validação de requisitos de sistema.

2) No processo de projeto de sistema, um protótipo pode ser usado para estudar soluções específicas do software e para apoiar o projeto de interface de usuário.

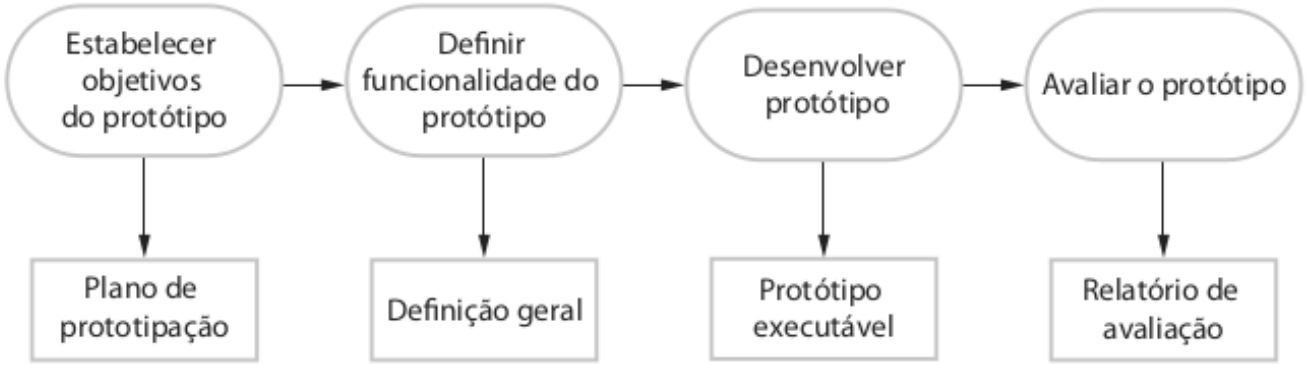
Prototipação

Fatec Araras

Protótipos do sistema permitem aos usuários ver quão bem o sistema dá suporte a seu trabalho. Eles podem obter novas ideias para requisitos e encontrar pontos fortes e fracos do software; podem, então, propor novos requisitos do sistema.

Prototipação

Figura 2.9 O processo de desenvolvimento de protótipo



Fonte: Sommerville

Prototipação

Fatec Araras

Os objetivos da prototipação devem ser explicitados desde o início do processo.

Estes podem ser o desenvolvimento de um sistema para prototipar a interface de usuário, o desenvolvimento de um sistema para validação dos requisitos funcionais do sistema ou o desenvolvimento de um sistema para demonstrar aos gerentes a viabilidade da aplicação.

Se os objetivos não são declarados, a gerência ou os usuários finais podem não entender a função do protótipo.

Prototipação

O próximo estágio do processo é decidir o que colocar e, talvez mais importante ainda, o que deixar de fora do sistema de protótipo.

Durante esse estágio, provisões devem ser feitas para o treinamento do usuário, e os objetivos do protótipo devem ser usados para derivar um plano de avaliação.

Prototipação

Fatec Araras

Um problema geral com a prototipação é que o protótipo pode não ser necessariamente usado da mesma forma como o sistema final.

O testador do protótipo pode não ser um usuário típico do sistema ou o tempo de treinamento durante a avaliação do protótipo pode ter sido insuficiente, por exemplo.

Protótipos não precisam ser executáveis para serem úteis.

Entrega Incremental

Fatec Araras

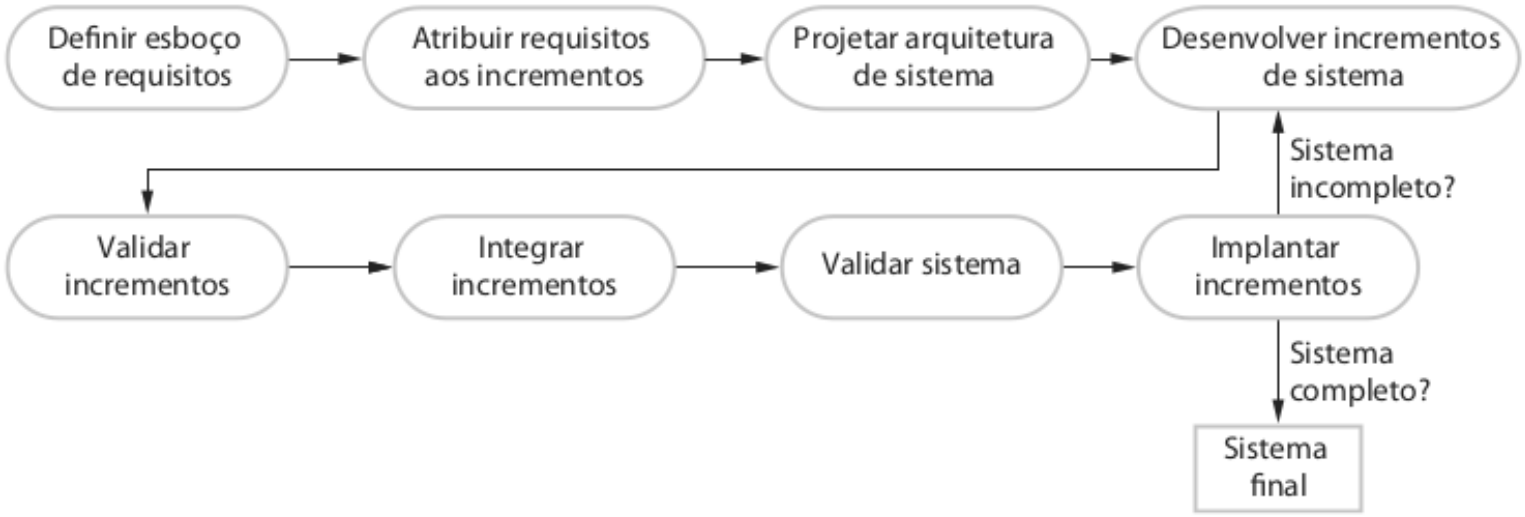
Entrega incremental é uma abordagem para desenvolvimento de software na qual alguns dos incrementos desenvolvidos são entregues ao cliente e implantados para uso em um ambiente operacional.

Em um processo de entrega incremental os clientes identificam, em linhas gerais, os serviços a serem fornecidos pelo sistema.

Entrega Incremental

Fatec Araras

Figura 2.10 Entrega incremental



Fonte: Sommerville

Entrega Incremental

Fatec Araras

Uma vez que os incrementos do sistema tenham sido identificados, os requisitos dos serviços a serem entregues no primeiro incremento são definidos em detalhes, e esse incremento é desenvolvido.

Durante o desenvolvimento, podem ocorrer mais análises de requisitos para incrementos posteriores, mas mudanças nos requisitos do incremento atual não são aceitas.

Entrega Incremental Vantagens

Fatec Araras

1. Os clientes podem usar os incrementos iniciais como protótipos e ganhar experiência, a qual informa seus requisitos para incrementos posteriores do sistema. Ao contrário de protótipos, trata-se, aqui, de partes do sistema real, ou seja, não existe a necessidade de reaprendizagem quando o sistema completo está disponível.

2. Os clientes não necessitam esperar até que todo o sistema seja entregue para obter ganhos a partir dele. O primeiro incremento satisfaz os requisitos mais críticos de maneira que eles possam usar o software imediatamente.

Entrega Incremental Vantagens

Fatec Araras

3. O processo mantém os benefícios do desenvolvimento incremental, o que deve facilitar a incorporação das mudanças no sistema.

4. Quanto maior a prioridade dos serviços entregues e, em seguida, incrementos integrados, os serviços mais importantes recebem a maioria dos testes. Isso significa que a probabilidade de os clientes encontrarem falhas de software nas partes mais importantes do sistema é menor.

Entrega Incremental Problemas

Fatec Araras

1. A maioria dos sistemas exige um conjunto de recursos básicos, usados por diferentes partes do sistema. Como os requisitos não são definidos em detalhes até que um incremento possa ser implementado, pode ser difícil identificar recursos comuns, necessários a todos os incrementos.

2. O desenvolvimento iterativo também pode ser difícil quando um sistema substituto está sendo desenvolvido. Usuários querem toda a funcionalidade do sistema antigo e, muitas vezes, ficam relutantes em experimentar um novo sistema incompleto. Portanto, é difícil obter feedbacks úteis dos clientes.

Entrega Incremental Problemas

Fatec Araras

3. A essência do processo iterativo é a especificação ser desenvolvida em conjunto com o software. Isso, contudo, causa conflitos com o modelo de compras de muitas organizações, em que a especificação completa do sistema é parte do contrato de desenvolvimento do sistema. Na abordagem incremental, não há especificação completa do sistema até que o último incremento seja especificado, o que requer uma nova forma de contrato, à qual os grandes clientes, como agências governamentais, podem achar difícil de se adaptar.

Entrega Incremental

Fatec Araras

Existem alguns tipos de sistema para os quais o desenvolvimento e a entrega incrementais não são a melhor abordagem. Esses sistemas são muito grandes, de modo que o desenvolvimento pode envolver equipes trabalhando em locais diferentes, além de alguns sistemas embutidos, em que o software depende do desenvolvimento de hardware, e de alguns sistemas críticos, em que todos os requisitos devem ser analisados na busca por interações capazes de comprometer a proteção ou a segurança do sistema.

Modelo espiral de Boehm

Modelo espiral de Boehm

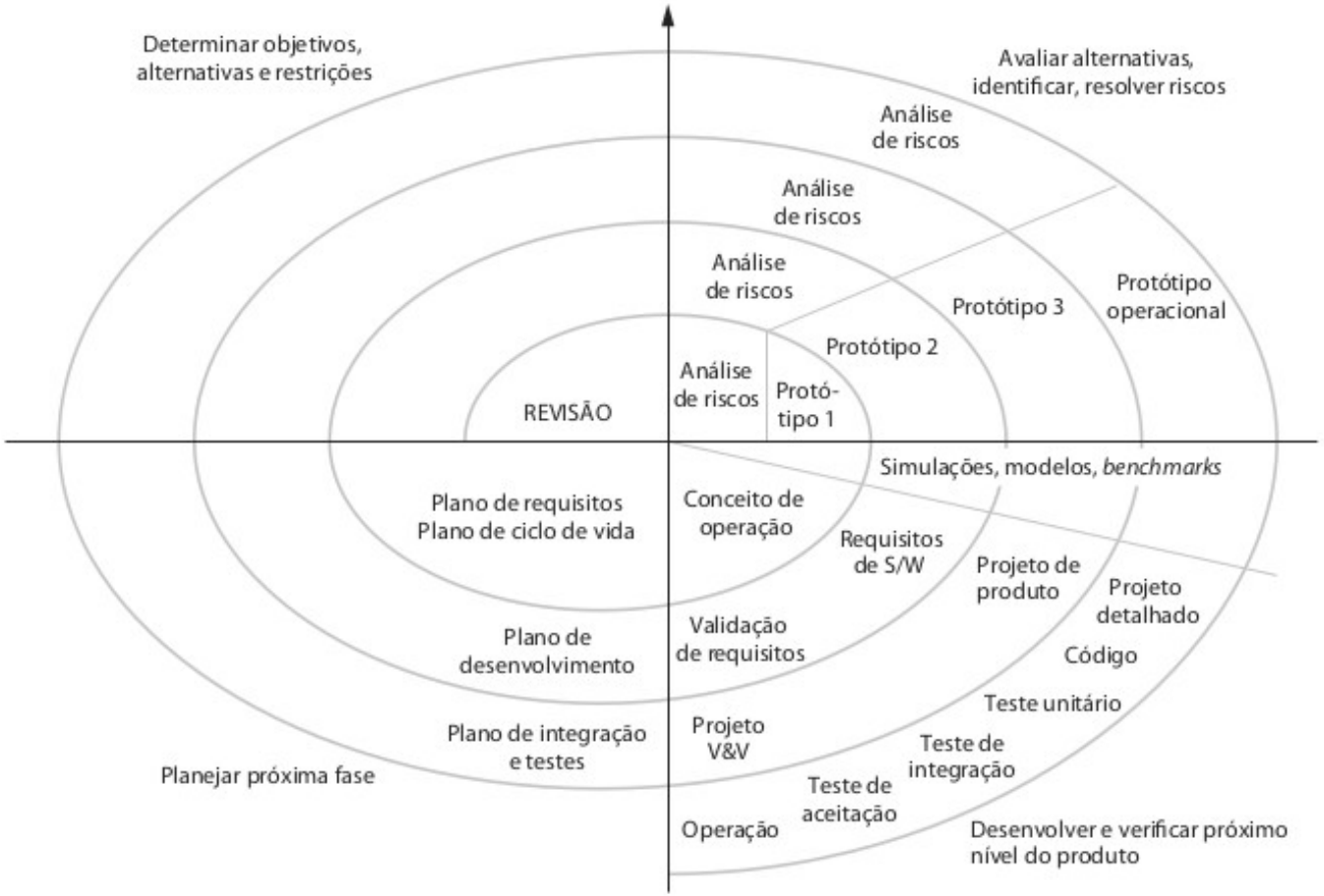
Framework de processo de software dirigido a riscos (o modelo em espiral)

O processo de software é representado como uma espiral, e não como uma sequência de atividades com alguns retornos de uma para outra.

Modelo espiral de Boehm

Fatec Araras

Figura 2.11 Modelo em espiral de processo de software de Boehm (©IEEE 1988)



Fonte: Sommerville

Modelo espiral de Boehm

Fatec Araras

Cada volta da espiral é dividida em quatro setores:

1. Definição de objetivos. Objetivos específicos para essa fase do projeto são definidos; restrições ao processo e ao produto são identificadas, e um plano de gerenciamento detalhado é elaborado; os riscos do projeto são identificados. Podem ser planejadas estratégias alternativas em função desses riscos.

2. Avaliação e redução de riscos. Para cada um dos riscos identificados do projeto, é feita uma análise detalhada. Medidas para redução do risco são tomadas. Por exemplo, se houver risco de os requisitos serem inadequados, um protótipo de sistema pode ser desenvolvido.

Modelo espiral de Boehm

3. Desenvolvimento e validação. Após a avaliação dos riscos, é selecionado um modelo de desenvolvimento para o sistema. Por exemplo, a prototipação descartável pode ser a melhor abordagem de desenvolvimento de interface de usuário se os riscos forem dominantes. Se os riscos de segurança forem a principal consideração, o desenvolvimento baseado em transformações formais pode ser o processo mais adequado, e assim por diante. Se o principal risco identificado for a integração de subsistemas, o modelo em cascata pode ser a melhor opção.

4. Planejamento. O projeto é revisado, e uma decisão é tomada a respeito da continuidade do modelo com mais uma volta da espiral. Caso se decida pela continuidade, planos são elaborados para a próxima fase do projeto.

Modelo espiral de Boehm

Fatec Araras

A principal diferença entre o modelo espiral e outros modelos de processo de software é seu reconhecimento explícito do risco.

Um ciclo da espiral começa com a definição de objetivos, como desempenho e funcionalidade.

Após a avaliação dos riscos, algum desenvolvimento é efetivado, seguido por uma atividade de planejamento para a próxima fase do processo. De maneira informal dizemos que o risco significa, simplesmente, algo que pode dar errado.

RUP
Rational Unified Process

RUP

IBM Rational Unified Process (RUP, 2003).

O RUP reconhece que os modelos de processo convencionais apresentam uma visão única do processo. Em contrapartida, o RUP é normalmente descrito em três perspectivas:

1. Uma perspectiva dinâmica, que mostra as fases do modelo ao longo do tempo.

2. Uma perspectiva estática, que mostra as atividades relizadas no processo.

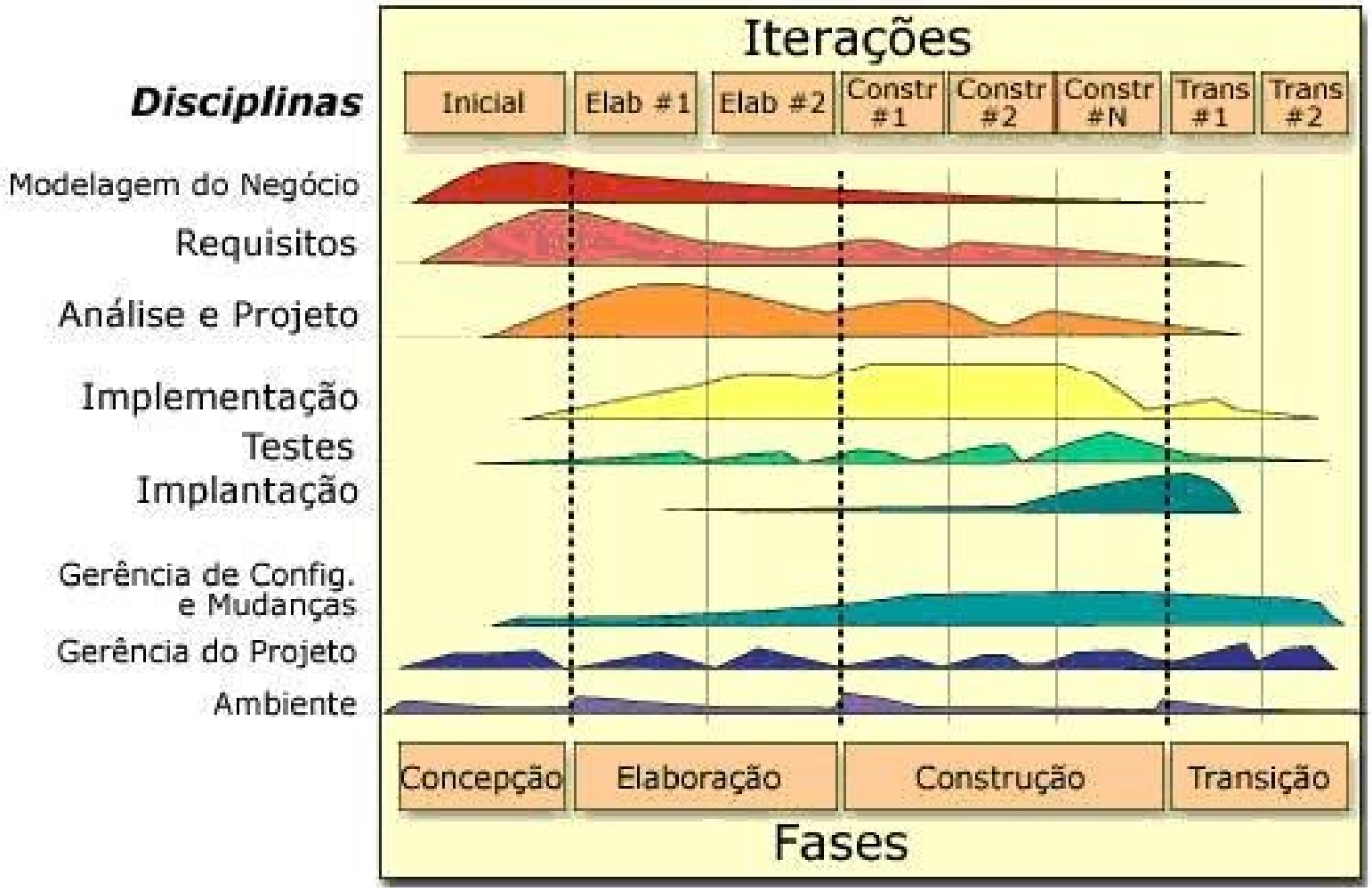
3. Uma perspectiva prática, que sugere boas práticas a serem usadas durante o processo.

RUP

O RUP é um modelo constituído de fases que identifica quatro fases distintas no processo de software.

No entanto, ao contrário do modelo em cascata, no qual as fases são equalizadas com as atividades do processo, as fases do RUP são estreitamente relacionadas ao negócio, e não a assuntos técnicos.

RUP


Iterações
Disciplinas
Inicial
Elab #1
Elab #2
Constr #1
Constr #2
Constr #N
Trans #1
Trans #2
Modelagem do Negócio
Requisitos
Análise e Projeto
Implementação
Testes
Implantação
Gerência de Config. e Mudanças
Gerência do Projeto
Ambiente
Concepção
Elaboração
Construção
Transição
Fases

RUP

Fatec Araras

1. **Concepção**. O objetivo da fase de concepção é estabelecer um *business case* para o sistema. Você deve identificar todas as entidades externas (pessoas e sistemas) que vão interagir com o sistema e definir as interações. Então, você deve usar essas informações para avaliar a contribuição do sistema para o negócio. Se essa contribuição for pequena, então o projeto poderá ser cancelado depois dessa fase.

2. **Elaboração**. As metas da fase de elaboração são desenvolver uma compreensão do problema dominante, estabelecer um framework da arquitetura para o sistema, desenvolver o plano do projeto e identificar os maiores riscos do projeto. No fim dessa fase, você deve ter um modelo de requisitos para o sistema, que pode ser um conjunto de casos de uso da UML, uma descrição da arquitetura ou um plano de desenvolvimento do software.

RUP

Fatec Araras

3. **Construção**. A fase de construção envolve projeto, programação e testes do sistema. Durante essa fase, as partes do sistema são desenvolvidas em paralelo e integradas. Na conclusão dessa fase, você deve ter um sistema de software já funcionando, bem como a documentação associada pronta para ser entregue aos usuários.

4. **Transição**. A fase final do RUP implica transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários e em seu funcionamento em um ambiente real. Isso é ignorado na maioria dos modelos de processo de software, mas é, de fato, uma atividade cara e, às vezes, problemática. Na conclusão dessa fase, você deve ter um sistema de software documentado e funcionando corretamente em seu ambiente operacional

RUP

Fatec Araras

Tabela 2.1 *Workflows* estáticos no Rational Unified Process

WORKFLOW	**DESCRIÇÃO**
Modelagem de negócios	Os processos de negócio são modelados por meio de casos de uso de negócios.
Requisitos	Atores que interagem com o sistema são identificados e casos de uso são desenvolvidos para modelar os requisitos do sistema.
Análise e projeto	Um modelo de projeto é criado e documentado com modelos de arquitetura, modelos de componentes, modelos de objetos e modelos de sequência.
Implementação	Os componentes do sistema são implementados e estruturados em subsistemas de implementação. A geração automática de código a partir de modelos de projeto ajuda a acelerar esse processo.
Teste	O teste é um processo iterativo que é feito em conjunto com a implementação. O teste do sistema segue a conclusão da implementação.
Implantação	Um *release* do produto é criado, distribuído aos usuários e instalado em seu local de trabalho.
Gerenciamento de configuração e mudanças	Esse *workflow* de apoio gerencia as mudanças do sistema (veja o Capítulo 25).
Gerenciamento de projeto	Esse *workflow* de apoio gerencia o desenvolvimento do sistema (veja os capítulos 22 e 23).
Meio ambiente	Esse *workflow* está relacionado com a disponibilização de ferramentas apropriadas para a equipe de desenvolvimento de software.

Fonte: Sommerville

Requisitos

Requisitos

Fatec Araras

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento.

O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado engenharia de requisitos (RE, do inglês *requirements engineering*).

Requisitos

1. **Requisitos de usuário** são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.

2. **Requisitos de sistema** são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software. O documento de requisitos do sistema (às vezes, chamado especificação funcional) deve definir exatamente o que deve ser implementado. Pode ser parte do contrato entre o comprador do sistema e os desenvolvedores de software.

Requisitos

Fatec Araras

Requisito de usuário

Definição de requisitos de usuário

1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

Requisito de sistema

Especificação de requisitos de sistema

1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.

1.2 Após 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.

1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.

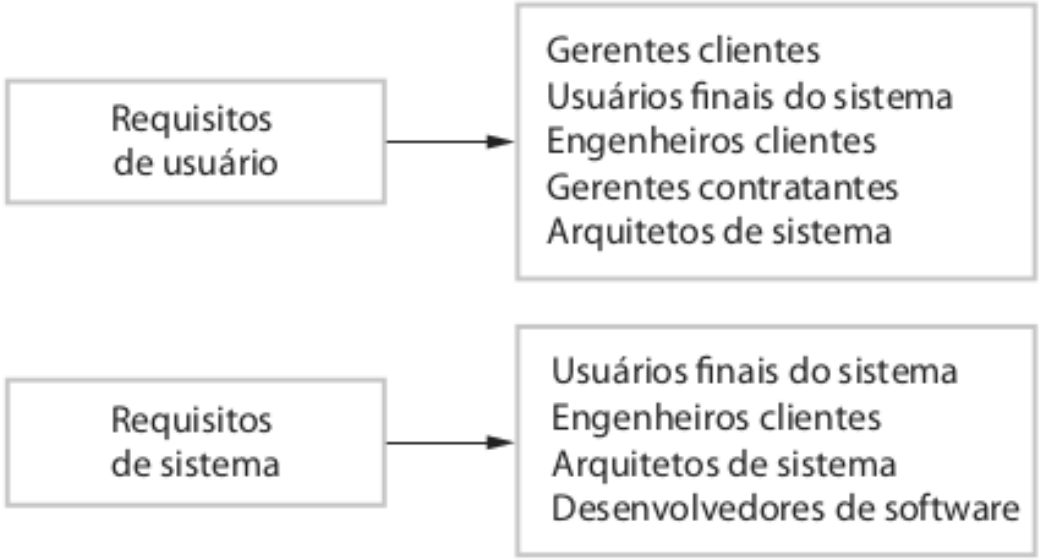
1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.

1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.

Fonte: Sommerville

Requisitos

Figura 4.2 Leitores de diferentes tipos de especificação de requisitos



Fonte: Sommerville

Requisitos
Funcionais e não funcionais

Fatec Araras

Os requisitos de software são frequentemente classificados como requisitos funcionais e requisitos não funcionais:

1. **Requisitos funcionais**. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

2. **Requisitos não funcionais**. São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo

Requisitos Funcionais

Fatec Araras

Os requisitos funcionais de um sistema descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos.

Requisitos funcionais do sistema variam de requisitos gerais, que abrangem o que o sistema deve fazer, até requisitos muito específicos, que refletem os sistemas e as formas de trabalho em uma organização.

A imprecisão na especificação de requisitos é a causa de muitos problemas da engenharia de software. É compreensível que um desenvolvedor de sistemas interprete um requisito ambíguo de uma maneira que simplifique sua implementação.

Requisitos Não Funcionais

Fatec Araras

Os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área.

Os requisitos não funcionais, como desempenho, proteção ou disponibilidade, normalmente especificam ou restringem as características do sistema como um todo. Requisitos não funcionais são frequentemente mais críticos que requisitos funcionais individuais.

Requisitos Não Funcionais

Fatec Araras

Os usuários do sistema podem, geralmente, encontrar maneiras de contornar uma função do sistema que realmente não atenda a suas necessidades. No entanto, deixar de atender a um requisito não funcional pode significar a inutilização de todo o sistema.

Por exemplo, se um sistema de aeronaves não cumprir seus requisitos de confiabilidade, não será certificado como um sistema seguro para operar; se um sistema de controle embutido não atender aos requisitos de desempenho, as funções de controle não funcionarão corretamente.

Requisitos Não Funcionais

Fatec Araras

Os requisitos não funcionais surgem por meio das necessidades dos usuários, devido a restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de software ou hardware, ou a partir de fatores externos, como regulamentos de segurança ou legislações de privacidade.

Requisitos Não Funcionais

Fatec Araras

1. Requisitos de produto. Esses requisitos especificam ou restringem o comportamento do software.

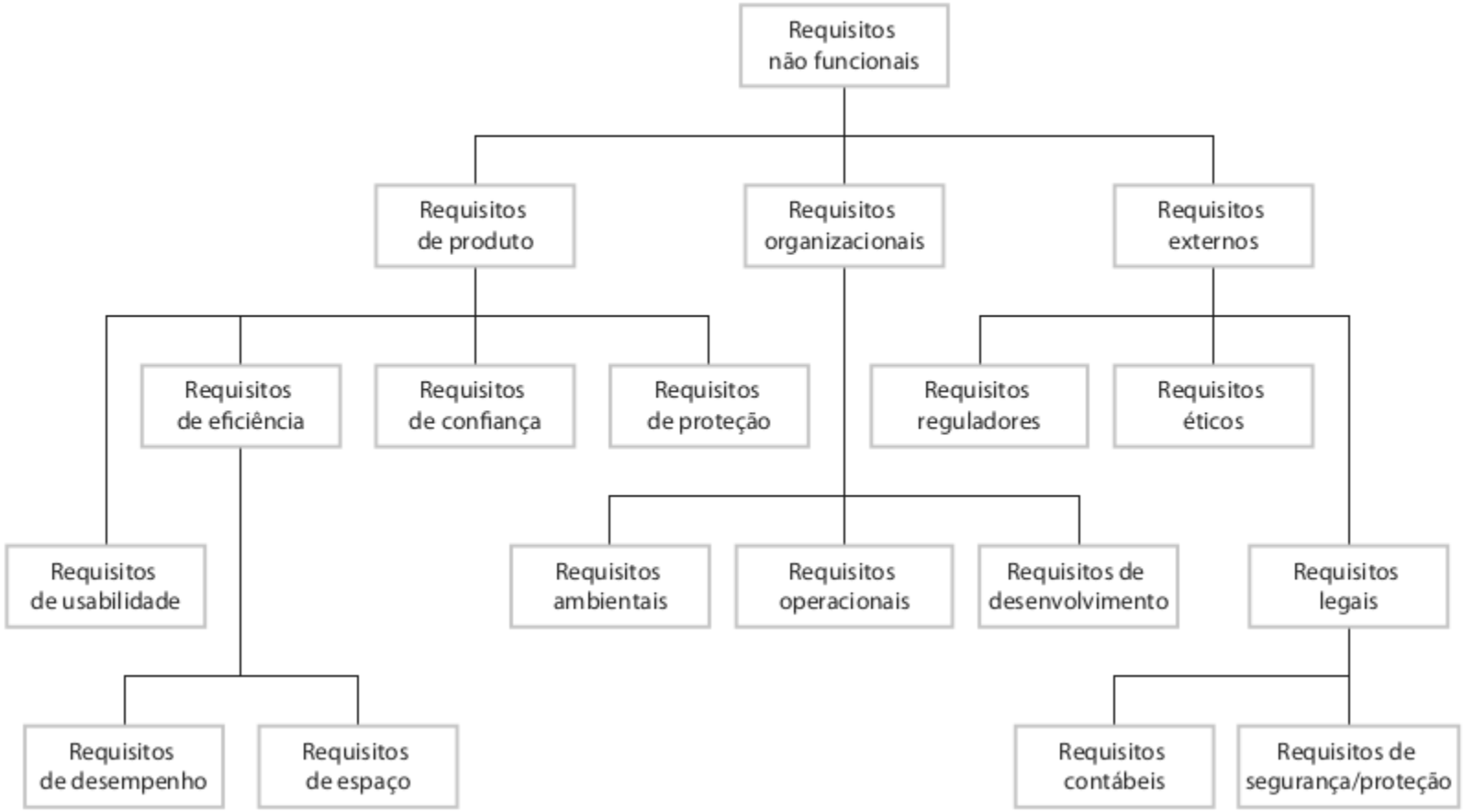
2. Requisitos organizacionais. Esses são os requisitos gerais de sistemas derivados das políticas e procedimentos da organização do cliente e do desenvolvedor.

3. Requisitos externos. Esse tipo abrange todos os requisitos que derivam de fatores externos ao sistema e seu processo de desenvolvimento.

Tipos de requisitos Não Funcionais

Fatec Araras

Figura 4.3 Tipos de requisitos não funcionais



Fonte: Sommerville

Tipos de requisitos Não Funcionais

Fatec Araras

Tabela 4.1 Métricas para especificar requisitos não funcionais.

Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização de tela
Tamanho	Megabytes Número de chips de memória ROM
Facilidade de uso	Tempo de treinamento Número de *frames* de ajuda
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Percentual de eventos que causam falhas Probabilidade de corrupção de dados em caso de falha
Portabilidade	Percentual de declarações dependentes do sistema-alvo Número de sistemas-alvo

Fonte: Sommerville

Processos de engenharia de requisitos

Processo de engenharia de requisito

Os processos de engenharia de requisitos podem incluir quatro atividades de alto nível.

Elas visam avaliar se o sistema é útil para a empresa (**estudo de viabilidade**), descobrindo requisitos (**elicitação e análise**), convertendo-os em alguma forma-padrão (**especificação**), e verificar se os requisitos realmente definem o sistema que o cliente quer (**validação**).

Processo de engenharia de requisito

Fatec Araras

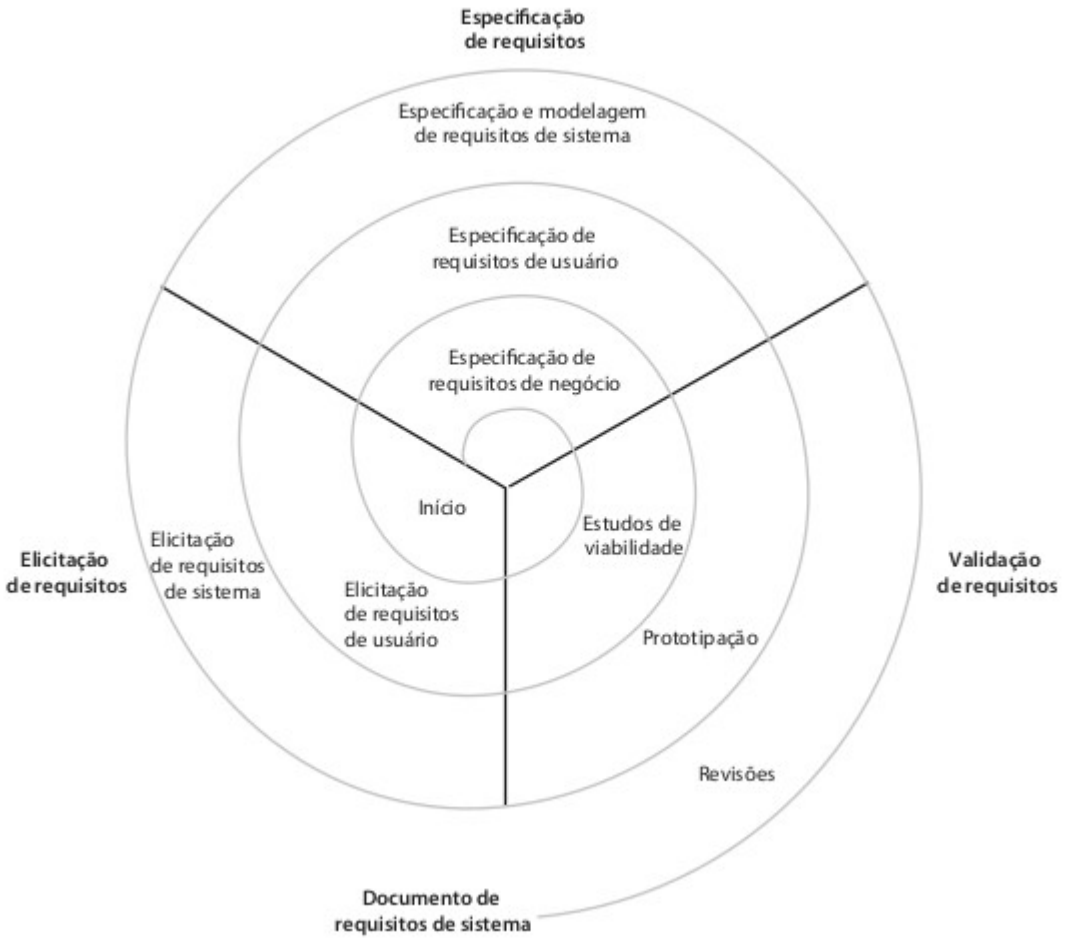
Algumas pessoas consideram a engenharia de requisitos o processo de aplicação de um método de análise estruturada, como a análise orientada a objetos (LARMAN, 2002).

Embora os métodos estruturados tenham um papel a desempenhar no processo de engenharia de requisitos, existe muito mais para a engenharia de requisitos do que o que é coberto por esses métodos.

Elicitação de requisitos, em particular, é uma atividade centrada em pessoas, e as pessoas não gostam de restrições impostas por modelos rígidos de sistema.

Processo de engenharia de requisito

Figura 4.5 Uma visão em espiral do processo de engenharia de requisitos.



Fonte: Sommerville

Elicitação e análise de requisitos

Fatec Araras

Após um estudo inicial de viabilidade, o próximo estágio do processo de engenharia de requisitos é a elicitação e análise de requisitos.

Nessa atividade, os engenheiros de software trabalham com clientes e usuários finais do sistema para obter informações sobre o domínio da aplicação.

Elicitação e análise de requisitos

Fatec Araras

A elicitação e análise de requisitos podem envolver diversos tipos de pessoas em uma organização.

Um *stakeholder* do sistema é quem tem alguma influência direta ou indireta sobre os requisitos do sistema.

Os *stakeholders* incluem os usuários finais que irão interagir com o sistema e qualquer outra pessoa em uma organização que será afetada por ele.

Elicitação e análise de requisitos

Fatec Araras

Cada organização terá sua própria versão ou instância desse modelo geral, dependendo de fatores locais, como a expertise do pessoal, o tipo de sistema a ser desenvolvido, as normas usadas, etc.

Elicitação e análise de requisitos

Fatec Araras

Figura 4.6 O processo de elicitação e análise de requisitos.

Fonte: Sommerville

Elicitação e análise de requisitos

Fatec Araras

1. Descoberta de requisitos. Essa é a atividade de interação com os stakeholders do sistema para descobrir seus requisitos.

2. Classificação e organização de requisitos. Essa atividade toma a coleção de requisitos não estruturados, agrupa requisitos relacionados e os organiza em grupos coerentes.

3. Priorização e negociação de requisitos.

4. Especificação de requisitos. Os requisitos são documentados e inseridos no próximo ciclo da espiral.

Descoberta de requisitos

Fatec Araras

Entrevistas formais ou informais com os stakeholders do sistema são parte da maioria dos processos de engenharia de requisitos.

1. Entrevistas fechadas, em que o stakeholder responde a um conjunto predefinido de perguntas.

2. Entrevistas abertas, em que não existe uma agenda predefinida.

Descoberta de requisitos

Fatec Araras

Cenários são descrições de exemplos de sessões de interação. Cada cenário geralmente cobre um pequeno número de interações possíveis.

A elicitação baseada em cenários envolve o trabalho com os stakeholders para identificar cenários e capturar detalhes que serão incluídos nesses cenários.

Os cenários podem ser escritos como texto, suplementados por diagramas, telas etc. Outra possibilidade é uma abordagem mais estruturada, em que cenários de eventos ou casos de uso podem ser usados.

Descoberta de requisitos

Em sua forma mais simples, um **caso de uso** identifica os atores envolvidos em uma interação e dá nome ao tipo de interação.

Essa é, então, suplementada por informações adicionais que descrevem a interação com o sistema.

Os casos de uso são uma técnica de descoberta de requisitos introduzida inicialmente no método Objectory (JACOBSON et al., 1993).

Descoberta de requisitos

Fatec Araras

Etnografia é uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos. Um analista faz uma imersão no ambiente de trabalho em que o sistema será usado.

O trabalho do dia a dia é observado e são feitas anotações sobre as tarefas reais em que os participantes estão envolvidos.

Validação de requisitos

Fatec Araras

A validação de requisitos é o processo pelo qual se verifica se os requisitos definem o sistema que o cliente realmente quer.

Ela se sobrepõe à análise, uma vez que está preocupada em encontrar problemas com os requisitos.

O custo para consertar um problema de requisitos por meio de uma mudança no sistema é geralmente muito maior do que o custo de consertar erros de projeto ou de codificação.

Gerenciamento de requisitos

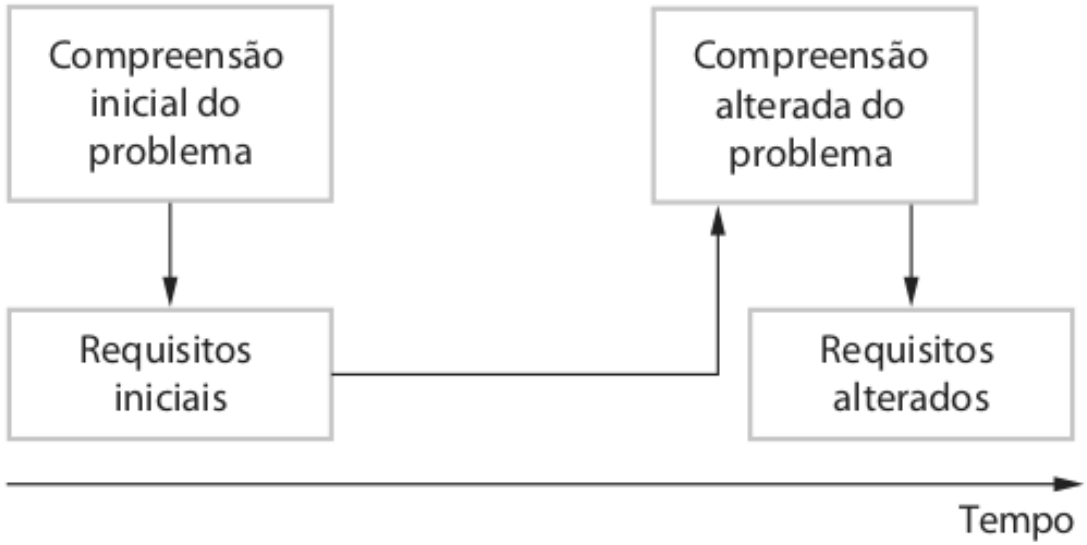
Fatec Araras

Durante o processo de software, o entendimento dos stakeholders a respeito do problema está em constante mutação. Logo, os requisitos de sistema devem evoluir para refletir essas novas percepções do problema.

Uma vez que um sistema tenha sido instalado e seja usado regularmente, inevitavelmente surgirão novos requisitos. É difícil para os usuários e clientes do sistema anteciparem os efeitos que o novo sistema terá sobre seus processos de negócio e sobre a forma que o trabalho é realizado.

Gerenciamento de requisitos

Figura 4.8 Evolução dos requisitos.



Fonte: Sommerville

Gerenciamento de requisitos

Fatec Araras

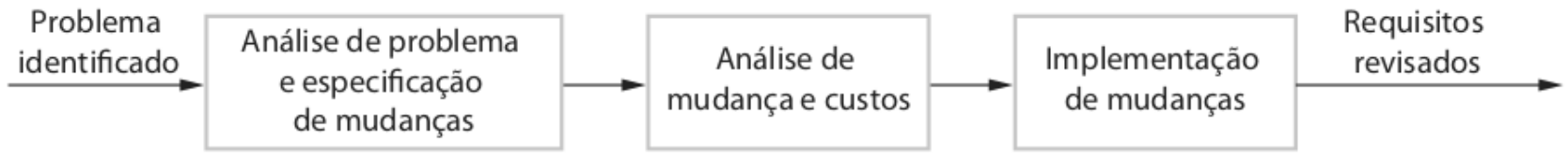
O gerenciamento de requisitos é o processo de compreensão e controle das mudanças nos requisitos do sistema. Você precisa se manter a par das necessidades individuais e manter as ligações entre as necessidades dependentes para conseguir avaliar o impacto das mudanças nos requisitos.

Você precisa estabelecer um processo formal para fazer propostas de mudanças e a ligação destas às exigências do sistema.

Gerenciamento de requisitos

Fatec Araras

Figura 4.9 Gerenciamento de mudança de requisitos.



Fonte: Sommerville

Gerenciamento de requisitos

Fatec Araras

Processos ágeis de desenvolvimento, como *Extreme Programming*, foram concebidos para lidar com requisitos mutáveis durante o processo de desenvolvimento. Nesses processos, quando um usuário propõe uma mudança nos requisitos, a mudança não passa por um processo formal de gerenciamento de mudanças. Pelo contrário, o usuário tem de priorizar essa mudança e, em caso de alta prioridade, decidir quais recursos do sistema planejados para a próxima iteração devem ser abandonados.

Fatec
Araras

Dúvidas

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

Pesquisa

Assistir ao vídeo

Desmistificando o Desenvolvimento Ágil

https://www.youtube.com/watch?v=FM2SS2Yjl1I&list=PLeKXYyZCJHxfeseMJ8JCE9J7qEjsWvoNI