

Processo de Engenharia de Requisitos

Centro de Informática - Universidade Federal de Pernambuco

Kiev Gama

kiev@cin.ufpe.br

Slides originais elaborados por Ian Sommerville e adaptado pelos professores Márcio Cornélio , Vinicius Garcia e Kiev Gama

O autor permite o uso e a modificação dos *slides* para fins didáticos



UNIVERSIDADE FEDERAL DE PERNAMBUCO

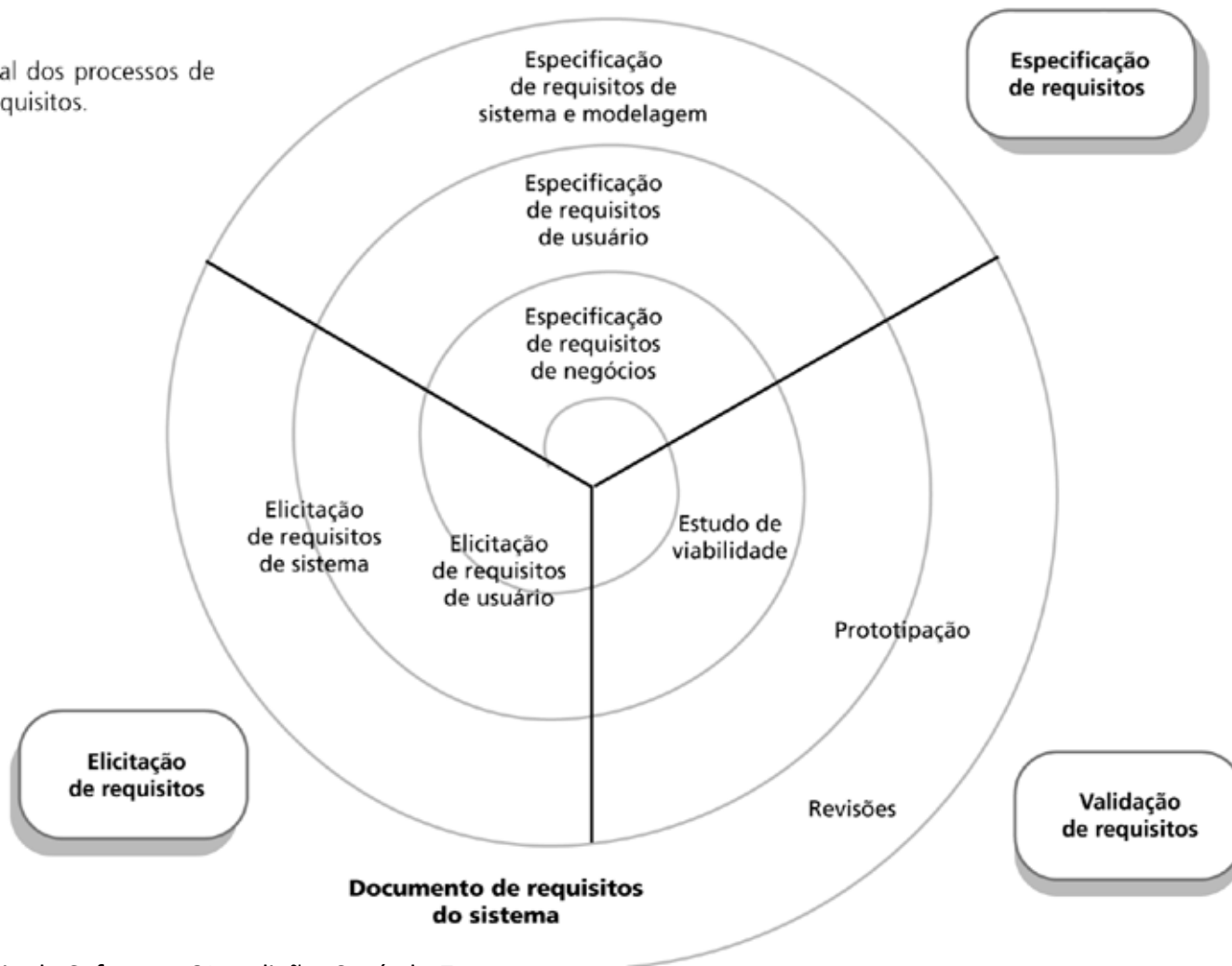
Processos de Engenharia de Requisitos

- Os requisitos e as formas de obtê-los e documentá-los variam drasticamente de um projeto para o outro
- Contudo, existe uma série de atividades genéricas **comuns** a todos os processos
 - Elicitação de requisitos;
 - Análise de requisitos;
 - Validação de requisitos;
 - Gerenciamento de requisitos.

Engenharia de requisitos

Figura 7.2

Modelo em espiral dos processos de engenharia de requisitos.



Elicitação e análise

- Envolve pessoal técnico trabalhando com os clientes para descobrir sobre o domínio da aplicação, os serviços que o sistema deve fornecer e sobre as restrições operacionais.
- Pode envolver
 - Usuários finais
 - Gerentes
 - Engenheiros envolvidos na manutenção
 - Especialistas de domínio
 - Representantes de sindicato, etc.
- Estes são chamados **stakeholders** (*partes interessadas*)

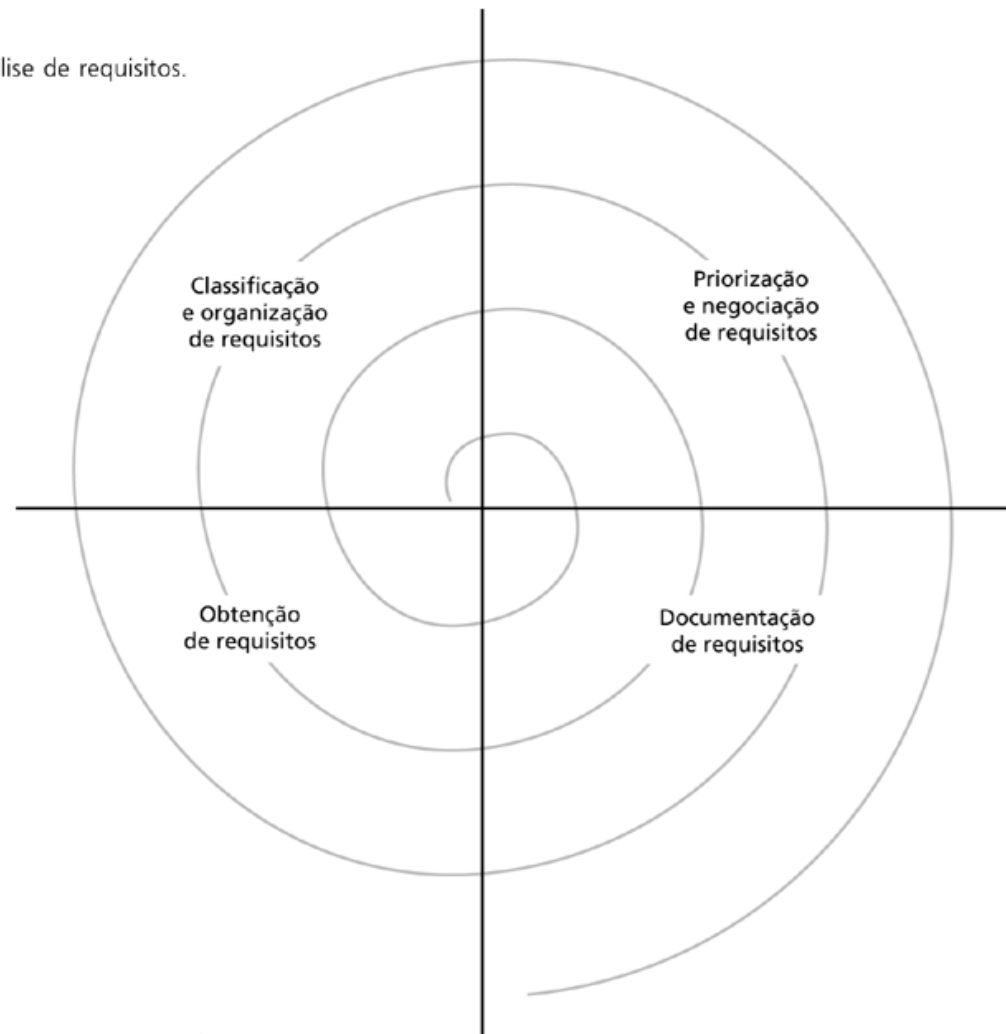
Problemas de análise de requisitos

- Stakeholders **não sabem** o que eles realmente querem.
- Stakeholders expressam requisitos em **seus próprios termos**.
- **Diferentes** stakeholders podem ter requisitos **conflitantes**.
- Fatores **organizacionais** e **políticos** podem influenciar os requisitos de sistema.
- **Mudanças** de requisitos **durante** o processo de análise

A espiral de requisitos

Figura 7.3

Processo de elicitação e análise de requisitos.



Atividades de processo

- Identificação (ou Elicitação) de requisitos
 - Interação com os stakeholders para coletar seus requisitos. Os requisitos de domínio são também descobertos neste estágio.
- Análise e Negociação de requisitos
 - Agrupa requisitos relacionados e organiza-os em conjuntos coerentes.
 - Priorização de requisitos e resolução de conflitos de requisitos.
- Documentação de requisitos
 - Os requisitos são documentados e colocados na próxima volta da espiral.

Identificação de requisitos

- Processo de reunir informações sobre os sistemas propostos e existentes
 - Obter requisitos de usuário e de sistema a partir dessas informações.
- As fontes de informação incluem documentação, stakeholders e as especificações de sistemas similares.
- Protótipos também podem ser usados tanto para descobrir quanto para validar requisitos
- Diferentes formas de fazer levantamento e análise de requisitos. Ex:
 - Orientado a pontos de vista
 - Cenários
 - Etnografia

Stakeholders de caixa eletrônico

- Clientes do banco
- Representantes de outros bancos
- Gerentes de bancos
- Caixas do banco
- Administradores de banco de dados
- Gerentes de proteção (segurança das informações)
- Departamento de marketing
- Engenheiros de manutenção de hardware e de software
- Reguladores de banco

Pontos de vista

- Maneira de estruturar os requisitos para representar as **perspectivas** de stakeholders diferentes.
 - Stakeholders podem ser classificados em diferentes pontos de vista.
- Essa análise de múltiplas perspectivas é importante, pois não há uma **maneira única** de analisar os requisitos

Tipos de pontos de vista

- **Pontos de vista de interação** são pessoas ou sistemas que interagem diretamente com o sistema.
 - Clientes e o banco de dados de contas são pontos de vista de interação.
- **Pontos de vista indiretos** são os stakeholders que não usam o sistema diretamente, mas afetam os requisitos.
 - Gerência, caixas do banco e pessoal de proteção são pontos de vista indiretos.
- **Pontos de vista de domínio** são as características e restrições de domínio que influenciam os requisitos.
 - Padrões para comunicações entre bancos representam pontos de vista de domínio

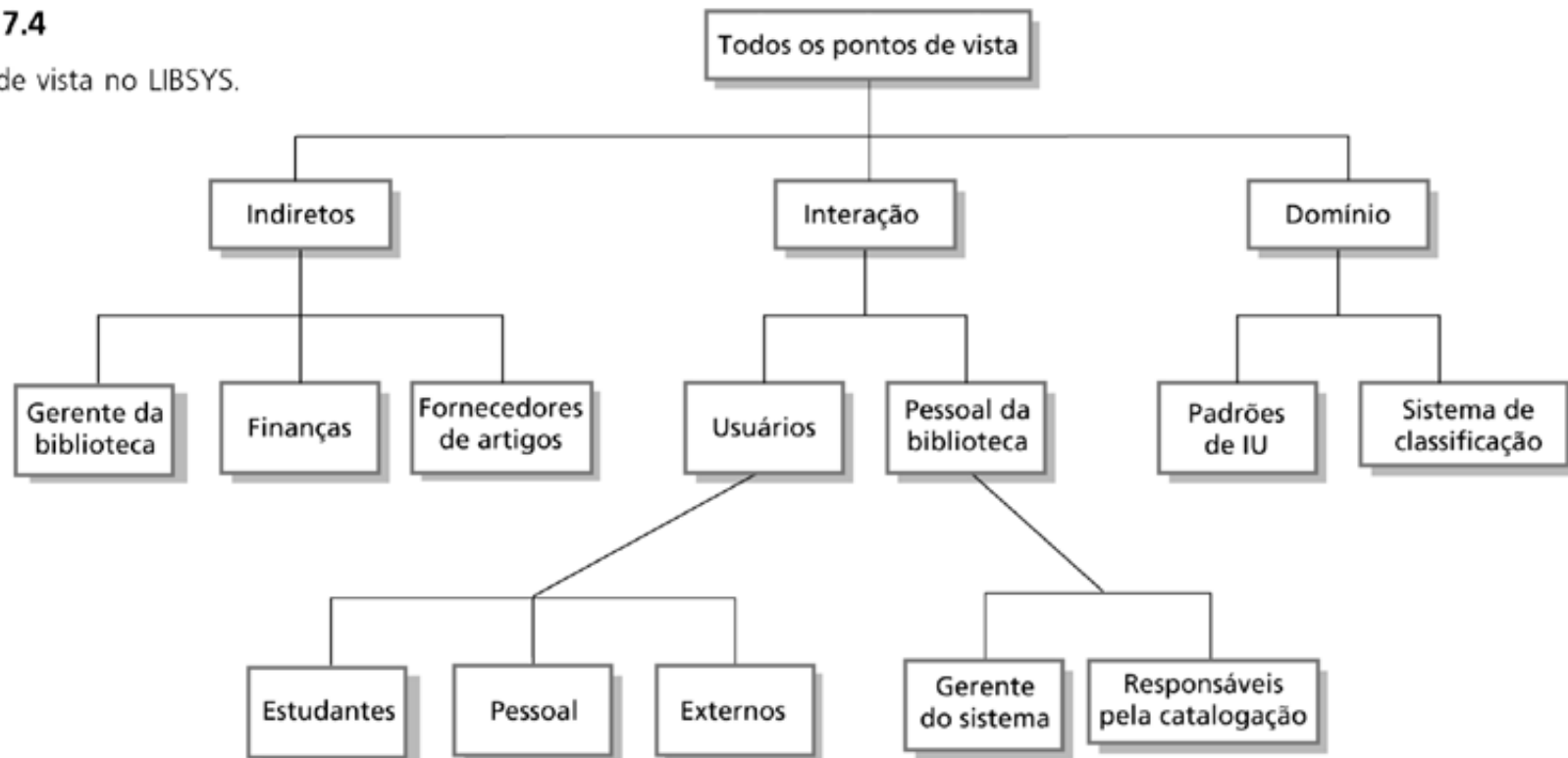
Identificação de pontos de vista

- **Identificar** pontos de vista usando:
 - Fornecedores e receptores de serviços do sistema;
 - Sistemas que devem interfacear diretamente com o sistema que está sendo especificado;
 - Regulamentos e padrões;
 - Fontes de requisitos de negócio e de requisitos não funcionais;
 - Engenheiros que têm de desenvolver e manter o sistema;
 - *Marketing* e outros pontos de vista de negócio.

Hierarquia de pontos de vista do LIBSYS

Figura 7.4

Pontos de vista no LIBSYS.



Entrevistas

- Em entrevista formal ou informal, a equipe de ER formula questões para os stakeholders sobre os sistemas que eles usam e o sistema a ser desenvolvido.
- Existem dois tipos de entrevistas
 - **Entrevistas fechadas**, onde um conjunto de questões predefinidas são respondidas.
 - **Entrevistas abertas**, onde não há um roteiro predefinido e onde uma variedade de assuntos são explorados com os stakeholders.

Entrevistas na prática

- Normalmente, uma mistura de entrevistas fechadas e abertas
- Entrevistas são boas para obtenção de um entendimento geral do que os stakeholders fazem e como eles podem interagir com o sistema.
- Entrevistas não são ideais para a compreensão de **requisitos de domínio**
 - Os engenheiros de requisitos podem não **entender a terminologia** específica de domínio;
 - Alguns conhecimentos de domínio são tão **específicos** que as pessoas acham difícil explicar ou pensam que não vale a pena mencioná-los

Cenários

- Cenários são **simulações** de como um sistema poderá ser usado
- Eles devem incluir
 - Uma descrição da situação inicial;
 - Uma descrição do fluxo normal de eventos;
 - Uma descrição do que pode dar errado;
 - Informação sobre outras atividades concorrentes;
 - Uma descrição do estado quando o cenário termina.
 - Para sistemas interativos, cenários funcionam bem em combinação com **protótipos da GUI**

Quadro 7.1

Cenário para download de artigo no LIBSYS.



Hipótese inicial: O usuário se conectou ao sistema LIBSYS e localizou a revista que contém a cópia do artigo.

Normal: O usuário seleciona o artigo a ser copiado. O sistema solicita que o usuário forneça as informações de assinante da revista ou indique uma forma de pagamento pelo artigo. O pagamento pode ser feito por meio de cartão de crédito ou com a informação de um número de conta da organização.

É solicitado, depois, que o usuário preencha um formulário de direitos autorais com os detalhes da transação e o envie ao sistema LIBSYS.

O formulário de direitos autorais é verificado e, caso aprovado, a versão do artigo em PDF é baixada na área de trabalho do LIBSYS no computador do usuário e este é avisado de que o artigo está disponível. É solicitado que o usuário selecione uma impressora, e uma cópia do artigo é impressa. Se o artigo estiver marcado como 'apenas para impressão', este será apagado do sistema do usuário após o término da impressão.

O que pode dar errado: O usuário pode não preencher o formulário de direitos autorais corretamente. Nesse caso, o formulário deverá ser reapresentado ao usuário para correção. Se o formulário reapresentado ainda estiver incorreto, a solicitação do usuário para o artigo será rejeitada.

O pagamento pode ser rejeitado pelo sistema; nesse caso, a solicitação do usuário para o artigo será rejeitada.

O download do artigo pode falhar, o que faz com que o sistema tente novamente até que a operação seja bem-sucedida ou que o usuário termine a sessão.

Pode não ser possível imprimir o artigo. Se o artigo não estiver marcado como 'apenas para impressão', ele será mantido na área de trabalho do LIBSYS. Caso contrário, o artigo será apagado e o custo do artigo será debitado na conta do usuário.

Outras atividades: Downloads simultâneos de outros artigos.

Estado de sistema após o término: O usuário estará conectado. O artigo baixado teria sido apagado da área de trabalho do LIBSYS caso estivesse marcado como 'apenas para impressão'.

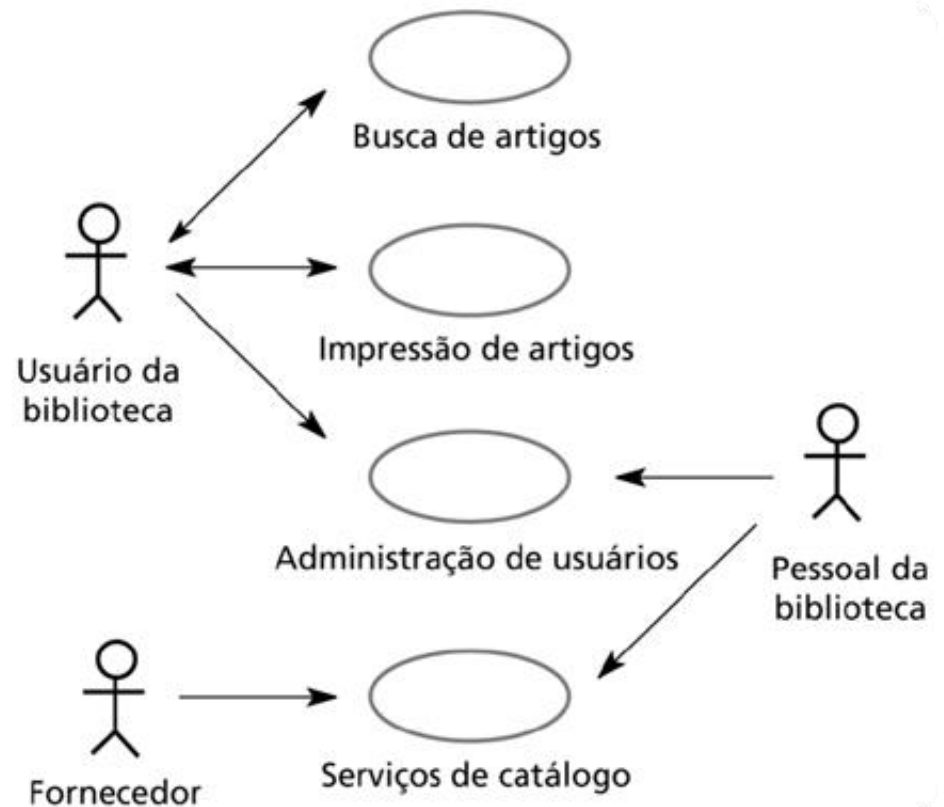
Casos de uso

- Os casos de uso constituem uma técnica baseada em cenários que identificam os agentes em uma interação e descrevem a interação em si.
 - Apoiados pela UML
 - **Diagramas de casos de uso** são usados para definir o escopo
 - **Especificações de casos de uso** são cenários como o descrito anteriormente
- Um conjunto de casos de uso deve descrever **todas as possíveis interações** com o sistema.

Alguns casos de uso do LIBSYS

Figura 7.6

Casos de uso para o sistema de biblioteca.



User stories

- Utilizado em metodologias ágeis (XP, Scrum) como definição de requisitos em alto nível
 - Na prática, detalha um item de backlog
 - São “lembretes” de conversas com stakeholders
- Formato simples, pequeno e objetivo
 - *Como um <tipo/papel de usuário>, eu quero <objetivo> para <razão/benefício>.*

Épicos

- Épicos: User stories “grandes”
 - Ex:
 - Como usuário quero fazer o backup do meu disco
 - Devem ser quebradas em várias histórias
 - Como usuário avançado, quero especificar arquivos e pastas para backup, de acordo com tamanho, data de criação e data de modificação
 - Como usuário, quero identificar pastas que não serão salvas para evitar que meu disco de backup fique cheio com coisas que não precisam ser salvas

Formato de User Stories

- Cartão ou *post-it* grande
- Frente
 - Texto da estória
 - ID
 - Prioridade
 - Esforço (em pontos)
- Verso
 - Testes de aceitação

Front of Card

173

As a student I want to purchase a parking pass so that I can drive to school

Priority: ~~High~~ Should
Estimate: 4

Back of Card

Confirmations:

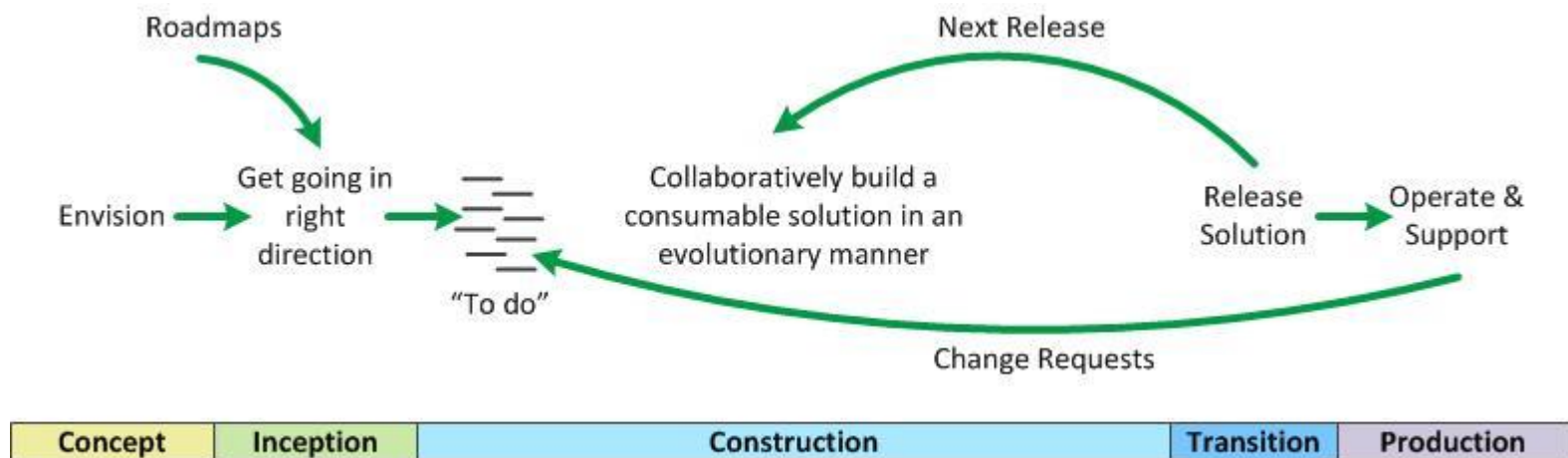
~~The student must pay the correct amount~~
One pass for one month is issued at a time
The student will not receive a pass if the payment isn't sufficient
The person buying the pass must be a currently enrolled student.
The student may only buy one pass per month.

User stories e planejamento do projeto

- Cronograma
 - O cronograma é uma lista de itens de trabalho
 - Itens de trabalho incluem histórias de usuário
 - Stakeholders podem incluir novas histórias ou (re)priorizá-las com a equipe
- Estimativa
 - Desenvolvedores estimam esforço para cada história
 - Atribuição de pontuação para cada história
 - Abordagem “poker”
 - Valores de uma escala não linear (Fibonnaci)

User stories no ciclo de vida do projeto

- **Iniciação:** Criação de estórias para identificar escopo do sistema
- **Construção:** Novas estórias serão identificadas, outras particionadas, (re)priorizadas ou mesmo removidas
- **Transição:** Possivelmente novas estórias podem ser identificadas nesta fase



Copyright 2014 Disciplined Agile Consortium

Detalhamento de user stories

- Estórias são incompletas, pois há poucas informações descritas em uma estória de usuário
- Em diferentes momentos é necessário desdobrá-la:
 - Conversas com stakeholders
 - Documentos auxiliares (esboços de protótipos, de telas, etc)
 - Planejamento da iteração
 - Quebrar em diferentes tarefas necessárias para implementar a estória
 - Durante a implementação da estória
 - Se necessário, criar esboços menos informais (diagrama de fluxo de dados, diagrama de sequência UML, etc)

Fatores sociais e organizacionais

- Sistemas de software são usados em um contexto social e organizacional. Isso pode influenciar, ou mesmo **dominar** os requisitos de sistema.
- Fatores sociais e organizacionais não são um ponto de vista único, mas são influências sobre todos os pontos de vista.
- É **muito difícil** saber se uma análise de fatores sociais e organizacionais está correta!

Etnografia

- Um analista despende um tempo considerável **observando e analisando** como as pessoas realmente trabalham.
- As pessoas não têm de explicar seu trabalho.
- Fatores sociais e organizacionais de importância podem ser observados.
- Estudos de etnografia têm mostrado que o trabalho é, geralmente, **mais rico e mais complexo** do que o sugerido pelos modelos simples de sistema.

Escopo da etnografia

- São requisitos originados a partir do modo como as pessoas **realmente trabalham**
 - Independem de como definições de processo sugerem que elas devam trabalhar.
- São requisitos originados a partir da **cooperação e da conscientização** das atividades de outras pessoas.

Mais Etnografia

- Etnografia funciona bem quando combinada com prototipação
 - O estudo etnográfico fornece **feedback** rápido sobre a aceitação e possíveis melhorias para um protótipo
- O desenvolvimento de protótipo resulta em questões não respondidas que tornam a análise etnográfica **mais focada**
- O problema com a etnografia é que ela estuda **práticas existentes** que podem ter alguma base histórica que não é mais relevante.
 - Não tão eficiente para descobrir **requisitos novos**

Validação de requisitos

- Dedicar-se a mostrar que os requisitos definem o **sistema que o cliente realmente deseja**.
- Custos de erros de requisitos são altos e, desse modo, a validação é muito importante
 - O custo da reparação de um erro de requisitos depois da entrega é muito maior que o custo de reparação de um erro de implementação

Verificação de requisitos

- **Verificação de validade.** O sistema fornece as funções que melhor apóiam as necessidades do cliente?
- **Verificação de consistência.** Existe algum tipo de conflito de requisitos? Para um mesmo requisito não pode haver contradição
- **Verificação de completude.** Todas as funções requisitadas pelo cliente foram incluídas?
- **Verificação de exequibilidade.** Os requisitos podem ser implementados com o orçamento e a tecnologia disponíveis?
- **Facilidade de verificação.** Os requisitos podem ser verificados? Usar conjunto de testes para demonstrar que a funcionalidade entregue atende o requisito

Técnicas de validação de requisitos

- Revisões de requisitos
 - Análise **manual** sistemática dos requisitos.
 - Potencialmente acompanhada por stakeholders
- Prototipação
 - Uso de um modelo executável do sistema para verificar requisitos
- Geração de casos de teste.
 - Desenvolvimento de testes para requisitos a fim de verificar a testabilidade
 - Testes de **aceitação**

Revisões de requisitos

- Revisões regulares devem ser feitas enquanto a definição de requisitos está sendo formulada.
- Ambos, **cliente** e **fornecedor**, devem ser envolvidos nas revisões.
- Revisões podem ser formais (com documentos completos) ou informais. Uma boa comunicação entre desenvolvedores, clientes e usuários pode resolver problemas nos estágios iniciais.

Revisão de requisitos

- **Facilidade de verificação.** O requisito é realisticamente testável?
- **Facilidade de compreensão.** O requisito é adequadamente compreendido?
- **Rastreabilidade.** A origem do requisito é claramente estabelecida?
- **Adaptabilidade.** O requisito pode ser mudado sem um grande impacto em outros requisitos?

Gerenciamento de requisitos

- Gerenciamento de requisitos é um processo para compreender e controlar as mudanças de requisitos
- Requisitos são, inevitavelmente, incompletos e inconsistentes
 - Novos requisitos surgem durante o processo inteiro
 - Os diferentes pontos de vista têm requisitos diferentes e estes são freqüentemente contraditórios.

Mudanças de requisitos

- Diferentes stakeholders atribuem diferentes prioridades para os mesmos requisitos
- Os clientes do sistema podem especificar os requisitos a partir de uma perspectiva de negócio que **conflita** com os requisitos do usuário final.
- Os ambientes técnico e de negócio do sistema mudam durante seu desenvolvimento
 - E frequentemente têm requisitos **diferentes**

Requisitos permanentes e voláteis

- **Requisitos permanentes.** São requisitos estáveis, derivados da atividade central da organização do cliente. Por exemplo, um hospital terá sempre médicos, enfermeiros, etc. Podem ser derivados dos modelos de domínio.
- **Requisitos voláteis.** São requisitos que mudam durante o desenvolvimento, ou quando o sistema estiver em operação. Um exemplo seria, em um hospital, os requisitos derivados da política de saúde.

Classificação de requisitos voláteis

Tabela 7.1 Classificação de requisitos voláteis

Tipo de requisito	Descrição
Requisitos mutáveis	Requisitos que mudam devido a mudanças no ambiente no qual a organização está operando. Por exemplo, em sistemas hospitalares, o financiamento do tratamento de pacientes pode mudar e, assim, exigir que informações de diferentes tratamentos sejam coletadas.
Requisitos emergentes	Requisitos que surgem à medida que a compreensão do sistema pelo cliente progride durante o desenvolvimento do sistema. O processo de projeto pode revelar novos requisitos emergentes.
Requisitos conseqüentes	Requisitos que resultam da introdução do sistema de computador. A introdução do sistema de computador pode mudar os processos da organização e criar novas formas de trabalho que geram novos requisitos de sistema.
Requisitos de compatibilidade	Requisitos que dependem de sistemas ou processos de negócios específicos dentro de uma organização. À medida que eles mudam, os requisitos de compatibilidade do sistema encomendado ou entregue podem também evoluir.

Rastreabilidade

- A rastreabilidade tem a ver com relacionamentos entre os requisitos, suas fontes e o **projeto** do sistema
 - É necessário manter essa informação registrada nos **locais apropriados**
- Rastreabilidade da fonte
 - Ligam requisitos aos **stakeholders** que os propuseram ou aos **elementos externos** que o criaram;
- Rastreabilidade de requisitos
 - É a ligação dos requisitos dependentes;
- Rastreabilidade de projeto
 - Ligações entre os requisitos e os módulos de projeto.

Gerenciamento de mudanças de requisitos

- Deve ser aplicado a **todas** as mudanças propostas aos requisitos
- Especialmente importante para sistemas **já prontos** ou em **estágios avançados de desenvolvimento**
- Estágios principais
 - **Análise de problema**: discutir problemas e mudanças de requisitos;
 - **Análise de mudança e estimativa de custo**: avaliar os efeitos das mudanças sobre outros requisitos;
 - **Implementação de mudança**: Modificar vários artefatos para refletir as mudanças.
- O impacto da mudança tem que ser avaliado para **TUDO O SISTEMA!**



How the customer explained it



How the Project Leader understood it



How the Business Consultant described it



How the Analyst designed it



How the Programmer wrote it



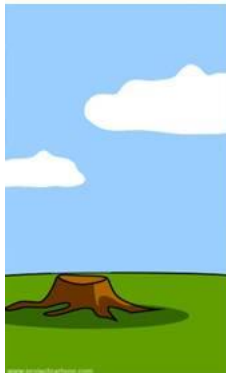
How the project was documented



What Operations installed



How it performed under load



How it was supported



What marketing advertised



How the customer was billed



What the customer really needed

Leituras recomendadas

- SOMMERVILLE, I. Engenharia de Software. 9ª. Ed. São Paulo: Pearson Education, 2011
– Capítulo 7
- AMBLER, S. Agile Modeling.
<http://www.agilemodeling.com/>