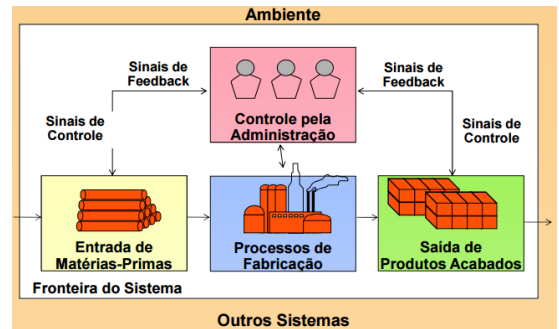


Resumo P1 de ISI

Aula 1 – Fundamentos

Desafios para empresas: projetar sistemas competitivos e eficazes, entender requisitos de sistema do ambiente de negócios global, criar arquitetura de informação que apoie os objetivos da organização, determinar o valor dos sistemas de informação para o negócio, projetar sistemas que as pessoas possam usar de maneira social e eticamente responsável.

Sistemas de informação: 5 áreas do conhecimento -> Desafios gerenciais, tecnologia da informação, conceitos básicos, processos de desenvolvimento e aplicações empresariais.



Sistema é um conjunto de partes coordenadas, que interagem para a realização de um conjunto de objetivos. Elementos a serem considerados: objetivos, ambiente, recursos, componentes, finalidades, medidas de rendimento e administração.

Diferentes tipos de sistema:

- De tempo real: Precisa da informação na hora, caso contrário pode causar danos (perda financeira, etc).
- De tempo real reativo: Sistema de controle metroviário, controle de paciente no hospital, controle aéreo (capta informações em tempo real e exige resposta rápida).

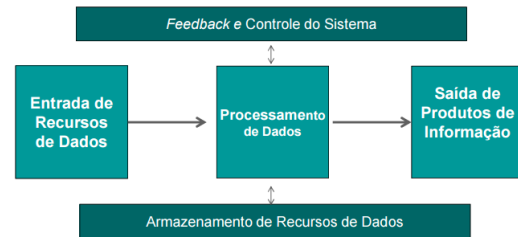
Sistemas de informação: Conjunto organizado de pessoas, hardware, software, redes de comunicação e recursos de dados que coleta, transforma e dissemina informações em uma organização. Não tem associado a ele características de perdas econômicas ou de vida (não é crítico). Pode ter partes críticas, mas não é de sua natureza. O principal objetivo é disseminar informações.

Componentes do SI:

- Recursos Humanos: Usuários finais (utilizam o sistema de informação ou a informação que ele produz) e especialistas em SI (desenvolvem e operam o sistema de informação). Stakeholders são todos os envolvidos com o desenvolvimento de um software (desenvolvedores, manutenção, usuário, etc).
- Recursos de Software: Programas e Procedimentos
- Recursos de Hardware: Máquinas e Mídias. Incluem todos os dispositivos físicos e materiais utilizados no processamento de informações, incluindo máquinas e mídias.
- Recursos de Rede: Meios de Comunicação e Suporte de Rede. Essenciais ao sucesso de operações de todos os tipos de organizações e de seus SIs computadorizados. Servidores, modems, roteadores, switches, antenas, cabeamento, internet, intranet, extranet...
- Recursos de Dados: Bancos de Dados (possuem e processam dados organizados) e Bases de Conhecimento (mantém o conhecimento em uma multiplicidade de formas tais como fatos e regras para a conclusão a respeito de um determinado assunto – áudio, vídeo, regras de negócio). Os dados constituem a matéria-prima dos sistemas de informação e estão entre seus recursos organizacionais mais valiosos.

Atividades Básicas de um SI:

- Operações de Entrada: Ação de capturar/coletar dados dentro da organização ou em seu ambiente externo. Podem ser tanto de forma manual quanto automatizada. Devem ser precisas para se atingir a(s) saída(s) desejada(s).



- Processamento: Ação de converter dados em forma significativa (informação). Pode envolver cálculos, comparações e tomadas de ações alternativas, e o armazenamento dos dados para uso futuro.

- Operações de Saída: Transferência da informação processada para pessoas ou atividades onde será usada. Envolve a produção de informações úteis geralmente na forma de documentos, relatórios e dados de transações.

- Feedback: Realimentação com dados sobre o desempenho do sistema. Saída utilizada para se fazer ajustes ou modificações nas atividades de entrada ou processamento. Erros ou problemas podem fazer com que os dados de entrada sejam corrigidos ou que um processo seja modificado. Garante que decisões possam ser tomadas em tempo hábil.

- Controle: Monitoramento e avaliação do feedback para determinar se o sistema irá atingir as suas metas. Realiza os ajustes necessários.

- Armazenamento: Atividade na qual os dados e as informações são guardados de forma organizada para uso posterior.

Dados: Fatos ou observações puras (em seu estado primário). São medidas objetivas dos atributos (características) de entidades como pessoas, lugares, coisas e eventos. Estrutura fundamental sobre a qual um SI é construído.

Informação: Fatos ou observações puras (em seu estado primário). São medidas objetivas dos atributos (características) de entidades como pessoas, lugares, coisas e eventos. Estrutura fundamental sobre a qual um SI é construído. Sua forma é agregada, manipulada e organizada. Seu conteúdo é analisado e avaliado. São colocados em um contexto adequado a um usuário humano. Dados com valor de conhecimento agregado. Possui propriedade econômica – o quão relevante é a informação para a empresa? Custo x Benefício. Vale gastar para desenvolver o SI? Qual o retorno?

Conhecimento: Capacidade de resolver problemas, inovar e aprender baseando-se em experiências prévias. Corpo ou regras, diretrizes e procedimentos utilizados para selecionar, organizar, manipular e compartilhar informação.

Ex: Dados do Aluno. Informação de cursos de cada aluno e evasão no ano. Conhecimento de “tal curso teve a maior taxa de evasão em 10 anos, com crescimento de 35%”.

Propriedades da informação: Precisa (isenta de erros), completa, econômica (relação entre custo e valor), flexível, confiável, relevante, simples, disponível e verificável (sua veracidade pode ser checada).

Organizações competem pelo domínio do conhecimento científico e tecnológico. Acumulando, processando, armazenando, acessando e disponibilizando informações por meio de redes de comunicação. Informação é um recurso estratégico para vantagem competitiva.

Aula 1 – Tipos de Sistemas de Informação

Como podemos utilizar a tecnologia da informação para alavancar a produtividade das pessoas e, consequentemente, das organizações? Alinhando os objetivos organizacionais com os sistemas de informação. SI automatiza trabalho manual e pode levar ao aumento da produtividade ou à dispersão.

Objetivos Organizacionais e os SIs: Sobrevivência; aumentar a eficiência das operações; aumentar a lucratividade; novos produtos, serviços e modelos de negócios; estreitar relacionamento com clientes e fornecedores; melhorar tomada de decisões (para isso, a informação deve ser precisa e atualizada); vantagem competitiva.

e-Business: É definido como o uso de tecnologias de Internet para interconectar e possibilitar processos de negócios, e-commerce, comunicação e colaboração dentro de uma organização e com seus clientes, fornecedores e demais envolvidos.

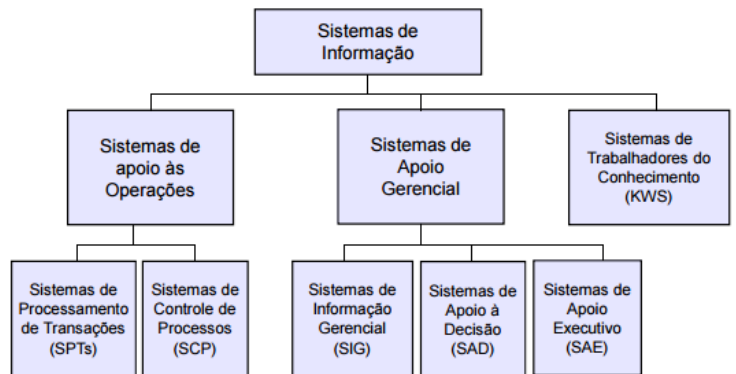
Tipos de SI:

- **Nível estratégico:** Determina objetivos de longo prazo, recursos e políticas. Decisões de efeito duradouro e mais difíceis de inverter.

- **Nível gerencial:** Ou tático, determina objetivos com prazos mais curtos, otimizando determinada área ou função empresarial e não a organização inteira. Impactos não tão incisivos à empresa, decisões menos relevantes.

- **Nível de conhecimento:** Avalia novas ideias para produtos e serviços, maneiras de comunicar novos conhecimentos e modos de distribuir informações. Cria a informação e o nível gerencial decide como resolver (passa para a camada de cima).

- **Nível operacional:** As decisões operacionais estão ligadas ao controle e às atividades operacionais da organização. Visam alcançar os padrões de funcionamento preestabelecidos. Dados rotineiros.



Tipos de SI:

- **Sistemas de Apoio às Operações:** Processar transações, controlar processos industriais, apoiar comunicação e colaboração, atualizar bases de dados.

- **Sistemas de Processamento de Transações (SPT):** Realizam e gravam as transações de rotina (diárias) necessárias para conduzir o negócio. No nível operacional, tarefas, recursos e objetivos são pré-definidos e altamente estruturados. Exemplos: Decisão para autorizar crédito a um cliente é feita por um supervisor de nível baixo de acordo com um critério pré-definido (decisão programada); decisão para autorizar a retirada de um livro é feita pelo bibliotecário de acordo com um critério (e.g. não está em débito com a biblioteca).

- **Sistemas de Controle de Processos (SCP):** Monitoram e controlam processos físicos contínuos. Computadores são utilizados para automaticamente tomar decisões que ajustam o processo físico de produção. Exemplos: Refinarias de petróleo, linhas de montagem de fábricas automatizadas, sistemas de expedição, etc.

- Sistemas de Apoio Gerencial: Fornecer informação e apoio para a tomada de decisão gerencial. Em geral, apoiar as necessidades de tomada de decisão estratégica e tática.

- Sistemas de Informação Gerencial (SIG): Suprem os gerentes com relatórios sobre o desempenho passado e presente da organização. Respondem: Como estamos indo? Normalmente, utilizam informações internas e não sobre eventos externos. Dependem dos SPTs (Sistemas de Processamento de Transações). Dados brutos obtidos dos SPTs. O que é realmente feito com os dados? Relatórios simples e repetitivos. Quais problemas são solucionados? Problemas repetitivos, rotineiros e bem estruturados.

- Sistemas de Apoio à Decisão (SAD): Ajudam os gerentes a tomarem decisões não usuais. Semi-estruturadas (parte do problema tem resposta clara). Não estruturadas (dependem de subjetividade). Propiciam aos gerentes criarem as informações que necessitam para decisões por meio de um sistema interativo de informação. O SAD utiliza informações dos SPT e SIG para auxiliar os processos de tomada de decisão. Exemplos: CRM (Customer Relationship Management). Auxilia a organização a gerenciar o cliente.

- Sistemas de Apoio Executivo (SAE): Fornecem informações críticas elaboradas especificamente para as necessidades de informação dos executivos. Apoio para o planejamento estratégico da organização. Ajudam as organizações a obterem um diferencial sobre os concorrentes. Exemplos: Análise de desempenho, ações dos concorrentes e desenvolvimento econômico. Podem alterar: Objetivos, operações, produtos, serviços ou relacionamentos ambientais da organização.

Trabalhadores do Conhecimento: Pessoas especializadas. Criam nova informação ou conhecimento. Exercem julgamento independente e com criatividade baseada no domínio de seu grande conhecimento especializado.

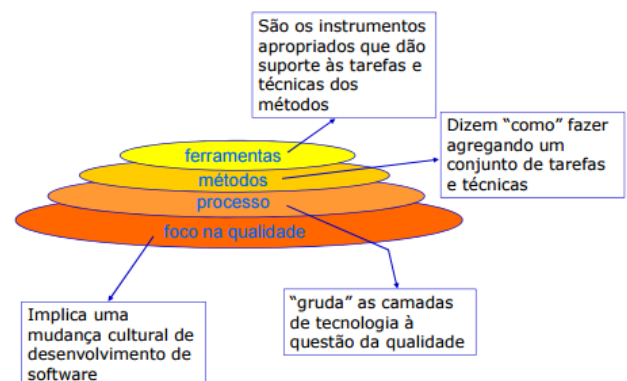
- Sistemas de Trabalhadores do Conhecimento (KWS): Ajudam os trabalhadores de conhecimento na criação e integração de novo conhecimento na organização. Incorporam mais ligações a dados e informações externas do que é habitual. Fácil acesso à base de conhecimentos externos armazenados eletronicamente. Ligações com profissionais de universidades e outras empresas. Exemplos: Sistemas de Projeto Apoiado por Computador (CAD), sistemas de realidade virtual (visualização, apresentação e simulação), Workstation de Investimento.

- Sistemas de Automação de Escritórios (OAS): Qualquer aplicação de tecnologia de informação que pretenda aumentar a produtividade dos trabalhadores da informação no escritório. Gerenciar de Documentos; Organizar Indivíduos e Grupos; Comunicar-se com Indivíduos e Grupos; Gerenciar Dados sobre Indivíduos e Grupos; Gerenciar Projetos.

Aula 2 – Processos de Desenvolvimento de Software

Três Fases de Desenvolvimento de Software: Definição (Engenharia do sistema, planejamento do projeto e análise de requisitos), Desenvolvimento (Projeto, geração do código e teste) e Manutenção.

Atividades de apoio: Controle e rastreamento de projeto, revisões técnicas formais, garantia de qualidade, gerenciamento de configuração, produção e preparação de documentos, gerenciamento de reusabilidade, medição e gerenciamento de risco.

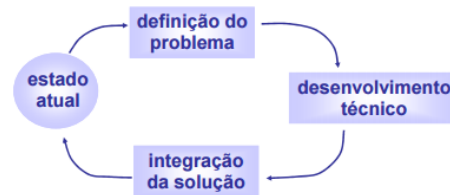


Produto de Software ≠ Processo de Software.

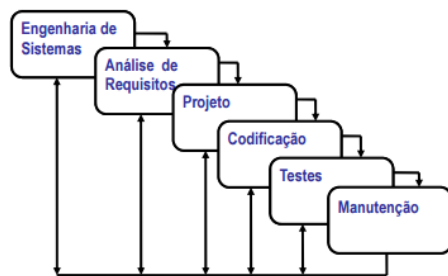
Manutenibilidade: Quão fácil é fazer a manutenção? Código fácil e claro, baixo acoplamento (alta coesão), formatação (identação), documentação interna e externa.

“Existe SI que não é software?” – Sim, manual.

Modelo de Processo de Desenvolvimento de Software:



Ciclo de Vida Clássico ou Modelo Cascata: Requer uma abordagem sistemática, sequencial ao desenvolvimento de software.



- Engenharia de Sistemas: Envolve a coleta de requisitos em nível do sistema, pequena quantidade de projeto e análise de alto nível. Visão essencial quando o software deve fazer interface com outros elementos (hardware, pessoas e banco de dados).

- Análise de Requisitos: Processo de coleta dos requisitos é intensificado e concentrado especificamente no software. Deve-se compreender o domínio da informação, a função,

desempenho e interfaces exigidos. Os requisitos (para o sistema e para o software) são documentados e revistos com o cliente.

- Projeto: Tradução dos requisitos do software para um conjunto de representações que podem ser avaliadas quanto à qualidade, antes que a codificação se inicie. Se concentra em 4 atributos do programa: Estrutura de Dados, Arquitetura de Software, Detalhes Procedimentais e Caracterização de Interfaces.

- Codificação: Tradução das representações do projeto para uma linguagem de programação, resultando em instruções executáveis pelo computador.

- Testes: Concentram-se nos aspectos lógicos internos do software, garantindo, por ex., que todas as instruções tenham sido testadas. Nos aspectos funcionais externos, para descobrir erros e garantir que a entrada definida produza resultados de acordo com o esperado.

- Manutenção: Em geral, o software sofre mudanças depois que é entregue ao cliente. Causas das mudanças: erros, adaptação do software para acomodar mudanças em seu ambiente externo e exigência do cliente para acréscimos funcionais e de desempenho.

- Problemas com o Modelo Cascata: Projetos reais raramente seguem o fluxo sequencial que o modelo propõe. Logo no início é difícil estabelecer explicitamente todos os requisitos. Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento. Muitas vezes os desenvolvedores ficam ociosos desnecessariamente, devido a estados bloqueadores (quando existem tarefas dependentes, membros da equipe devem aguardar que outros terminem).

Prototipação: Processo que possibilita que o desenvolvedor crie um modelo do software que deve ser construído. Idealmente, o modelo (protótipo) serve como um mecanismo para identificar os requisitos de software. Apropriado para quando o cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento e saída com detalhes. Identificados os requisitos, o protótipo deve ser descartado/aproveitado, para que a versão de produção possa ser construída/aprimorada considerando os critérios de qualidade. *Além de modelo de ciclo de vidas, pode ser considerado uma técnica de levantamento de requisitos.*

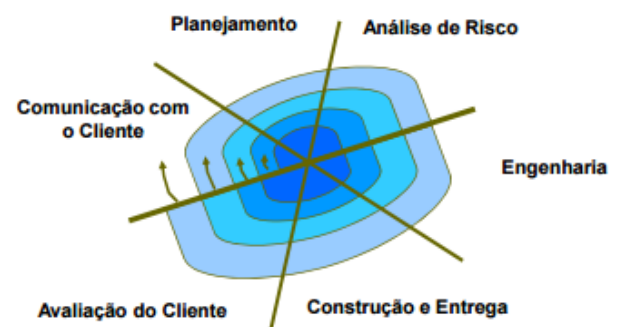


- Tipos de protótipos: Evolutivo – o produto final vai sendo desenvolvido à medida que os protótipos vão sendo construídos; no final, basta acertarem-se alguns detalhes. Descartável – ao final dos ciclos de protótipos, o produto final ainda vai ser construído.

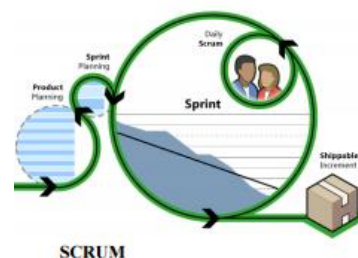
- Problemas com a Prototipação: Cliente não sabe que o software que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo. Cliente não aceita bem a idéia de que a versão final do software vai ser construída e "força" a utilização do protótipo como produto final. Desenvolvedor frequentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo.

Modelos de Processo Evolutivos: São iterativos e permitem um desenvolvimento incremental, de forma que a cada incremento uma versão mais completa do software é liberada. Assim, uma versão limitada pode ser introduzida para atender à competitividade e pressões do negócio. Os requisitos do produto e do negócio vão evoluindo ao longo das iterações. Engloba a natureza iterativa da Prototipação com os aspectos sistemáticos e controlados do Modelo Cascata.

- Modelo Espiral: Tem como diferencial a análise de risco formal e detalhada. Custo vs. Benefício: Vale a pena melhorar o software? O público vai querer migrar de versão?



- Métodos Ágeis: Necessidade de entrega rápida e flexível. Priorizam: Interação e pessoas ao invés de procedimentos e ferramentas; software funcional ao invés de documentação extensa; colaboração com o cliente ao invés de negociações de contrato; atendimento a mudanças ao invés de seguir um plano rígido. Trabalha com iterações de tempos iguais. Melhor para aplicações menos complexas e equipes menores. Toda a equipe sabe o que está acontecendo e pode trabalhar em qualquer parte do projeto. eXtreme Programming (XP) e Scrum. O Scrum



determina estrutura de trabalho, é só um framework de trabalho, pois não define as tarefas. Posso ter o XP enquanto uso o Scrum, por exemplo.

Aula 3 – Engenharia de Requisitos

Um requisito de software descreve o que é requerido para que o software cumpra seu papel.

Requisitos do Usuário: Declarações em linguagem natural mais diagramas dos serviços fornecidos pelo sistema e suas restrições operacionais. Escrito para os clientes.

Requisitos do Sistema: Um documento estruturado mostrando descrições detalhadas dos serviços do sistema. Escrito como um contrato entre cliente e contratado.

Requisitos Funcionais: Funcionalidade Necessária.

Requisitos Não Funcionais: Propriedade, Restrição. Uso de métricas.

Requisitos do Domínio: São próprios do domínio da aplicação e que refletem características desse domínio. Podem ser requisitos funcionais ou não-funcionais.

Os requisitos podem ser definidos operacionalmente, usando uma linguagem do tipo de uma linguagem de programação (PDL), mas com mais flexibilidade de expressão. Seu uso é mais apropriado em duas situações: Quando uma operação é especificada como uma sequência de ações em que a ordem é importante; Quando interfaces de hardware e software devem ser especificados.

Os Requisitos do Usuário devem ser escritos em Linguagem Natural, Tabelas e Diagramas. Os Requisitos do Sistema têm por objetivo explicar as funções que o sistema deve fornecer. Os Requisitos do Sistema podem ser escritos em Linguagem Natural, PDL, Linguagem Formal, etc. O Documento de Requisitos do software é uma declaração de concordância em relação aos requisitos definidos para o sistema.

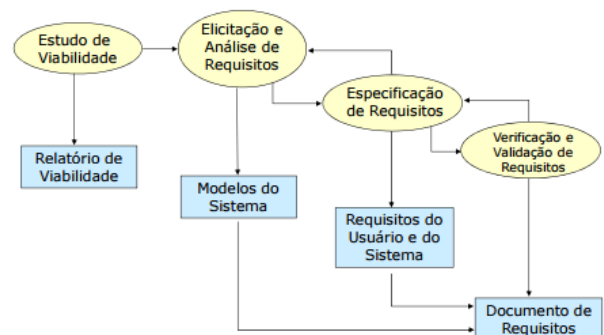
Técnicas de Elicitação de Requisitos: Também chamadas de técnicas de comunicação, são propostas visando auxiliar a comunicação e a extração dos requisitos. As mais utilizadas são: Entrevistas, Questionários, Cenários, Histórias do usuário, Brainstorming.

Processo de Engenharia de Requisitos: É o processo usado para descobrir, analisar e validar os requisitos do sistema. Envolve todas as atividades necessárias para criar e manter o Documento de Requisitos.

- Estudo de Viabilidade: O sistema contribui para os objetivos gerais da organização? O sistema pode ser construído usando a tecnologia atual e dentro do orçamento? O sistema pode ser integrado com outros sistemas já em operação? É necessária nova tecnologia?

Quais as habilidades? Quais facilidades devem ser supridas pelo sistema proposto? Quais são os problemas com os processos atuais?

- Elicitação e Análise dos Requisitos: Visa esclarecer sobre o domínio da aplicação, os serviços que o sistema deve prover e as restrições operacionais do sistema. Envolve os stakeholders. Diferentes stakeholders podem ter requisitos conflitantes. Os fatores organizacionais e políticos podem influenciar os requisitos. Os requisitos mudam durante o processo de análise. A Análise de Requisitos pode envolver três atividades de estruturação que podem resultar em diferentes modelos: Particionamento



(Identifica o relacionamento estrutural entre as entidades), Abstração (Identifica generalidades entre a entidades), Projeção (Identifica diferentes maneiras de olhar o problema).

- Verificação e Validação de Requisitos: Tem por objetivo mostrar que os requisitos realmente definem o sistema que o cliente deseja. Tem muito em comum com a Verificação de Requisitos da fase de Elicitação e Análise, pois se preocupa em descobrir problemas nos requisitos. A diferença é que na fase de Análise trabalha-se com requisitos incompletos e na Validação trabalha-se com o Documento de Requisitos. Os custos associados com requisitos errados são extremamente altos, o que faz da Validação uma atividade muito importante.

- Verificações a fazer: Validade, consistência, completeza, realismo, facilidade de verificação.

- Técnicas de Validação de Requisitos: Revisão, prototipação e geração de casos de teste

- As Revisões podem ser Formais (com o preenchimento de documentos) ou Informais. Itens a verificar: facilidade de verificação, facilidade de compreensão, facilidade de rastreamento, facilidade de adaptação.

Gerenciamento de Requisitos: É o processo de gerenciar as mudanças nos Requisitos durante o Processo de Engenharia de Requisitos e o desenvolvimento do sistema.

- Requisitos Permanentes: São estáveis e derivados da atividade fundamental da organização do cliente. Podem ser derivados dos modelos do Domínio. Ex: um hospital sempre terá médicos, enfermeiras, etc.

- Requisitos Voláteis: mudam durante o desenvolvimento ou quando o sistema está em uso. Ex: políticas sobre assistência médica.

- Mutáveis: Mudam em decorrência das mudanças do ambiente em que o sistema está operando.

- Emergentes: Surgem à medida que a compreensão do sistema aumenta.

- Consequentes: Resultam da própria introdução do sistema, o qual pode mudar a rotina de trabalho.

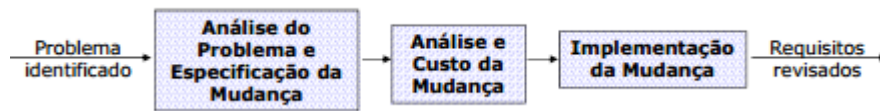
- De Compatibilidade: Dependem de outros sistemas ou processos de negócio específicos da organização.

Rastreabilidade: É a facilidade com que se consegue identificar o relacionamento entre os requisitos, suas fontes e o projeto do sistema. Seus tipos são: Rastreabilidade da Origem (Links que se criam entre os requisitos e os stakeholders que propuseram os requisitos), Rastreabilidade dos Requisitos (Links entre requisitos dependentes), Rastreabilidade de Projeto (Links dos requisitos para o projeto). *Por que ter matriz de rastreabilidade? Se eu alterar um requisito e ele tem relacionamento com outro, preciso alterar esse outro.*

Quais requisitos não funcionais automaticamente podem ser melhorados ou contemplados com a matriz de rastreabilidade (tabela de requisito vs requisito)? Manutenibilidade e teste de regressão. Teste que tem que ser feito quando o software está operacional para o cliente... Minha mudança provocou efeitos colaterais? Quais? Não testa o produto inteiro, só as partes impactadas pela manutenção.

Req. id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		U	R					
1.2			U			R		U
1.3	R			R				
2.1			R		U			U
2.2								U
2.3		R		U				
3.1								R
3.2							R	

R - relacionamento fraco
U - relacionamento forte



Documento de Requisitos: Declaração oficial do “o que” que é solicitado aos desenvolvedores do sistema. O DR deve incluir tanto a definição (requisitos do usuário) como a especificação (requisitos do sistema) dos requisitos. O DR NÃO é um documento de projeto. Na medida do possível, ele deve declarar “O QUE” o sistema deve fazer e não “COMO” deve ser feito.

Padrão IEEE: Introdução, Descrição Geral, Requisitos específicos, Apêndices, Índice.

Aula 4 – Modelos de Caso de Uso

Modelo de casos de uso: Representação das funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema que interagem com o mesmo. Representa os requisitos funcionais do sistema. Composto de duas partes, uma textual, e outra gráfica.

Casos de uso: Especificação de uma sequência de interações entre um sistema e os agentes externos. Define parte da funcionalidade de um sistema, sem revelar a estrutura e o comportamento interno desse sistema.

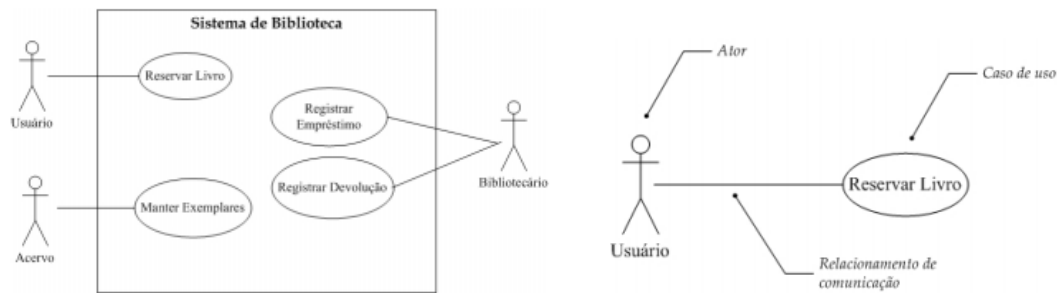
Exemplo de descrição tabular

Cliente	Sistema
Insere seu cartão no caixa eletrônico.	Apresenta solicitação de senha.
Digita senha.	Valida senha e exibe menu de operações disponíveis.
Solicita realização de saque.	Requisita quantia a ser sacada.
Fornece o valor da quantia que deseja sacar.	Fornece a quantia desejada e imprime o recibo para o Cliente
Retira a quantia e o recibo.	

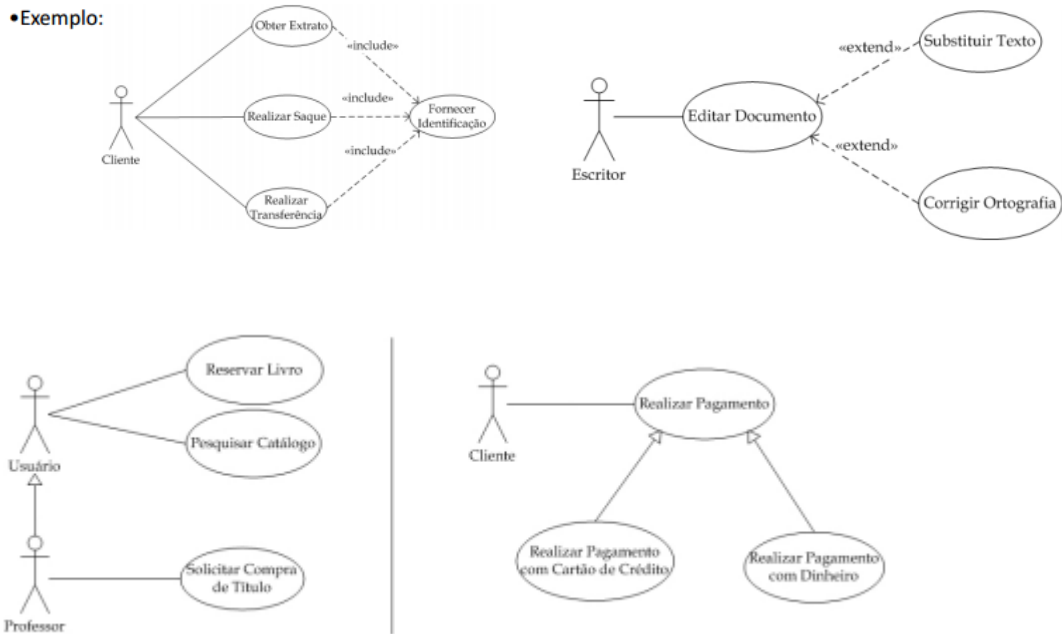
Se a narrativa era válida 100 anos atrás e continuará válida daqui a 100 anos, então o caso de uso é essencial. Caso contrário, trata-se de um caso de uso real.

Atores: Elemento externo que interage com o sistema. Um ator corresponde a um papel representado em relação ao sistema. O nome dado a um ator deve lembrar o seu papel, em vez de lembrar quem o representa. Representa um conjunto coerente de papéis que os usuários de casos desempenham quando interagem com o sistema. Um caso de uso representa o que um ator quer que o sistema faça.

Diagrama de casos de uso (DCU): Representa graficamente os atores, casos de uso e relacionamentos entre os elementos. Tem o objetivo de ilustrar em um nível alto de abstração quais elementos externos interagem com que funcionalidades do sistema. Uma espécie de “diagrama de contexto”. Apresenta os elementos externos de um sistema e as maneiras segundo as quais eles as utilizam. Os elementos mais comuns em um DCU são: Ator e Caso de uso. Além disso, a UML define diversos de relacionamentos entre esses elementos para serem usados no modelo de casos de uso: Comunicação, Inclusão, Extensão e Generalização.



•Exemplo:

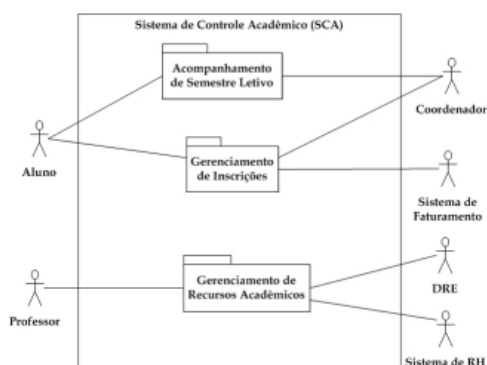


Se X ---- <<include>>----->W: X depende de W para acontecer. X chama obrigatoriamente W.

Se W -----<<extend>>----->X: X pode eventualmente chamar W.

Caso de Uso Primário: Representa os objetivos dos atores.

Caso de Uso Secundário: Aquele que não traz benefício direto para os atores, mas que é necessário para que sistema funcione adequadamente.



Uma breve descrição para cada ator deve ser adicionada ao MCU. O nome de um ator deve lembrar o papel desempenhado pelo mesmo. Exemplo: “Aluno: representa pessoas que fazem um curso dentro da universidade.”.

Documentação dos casos de uso

- | | |
|----------------------|--------------------------|
| ▣ Nome | ▣ Fluxo Principal |
| ▣ Descrição | ▣ Fluxos Alternativos |
| ▣ Identificador | ▣ Fluxos de Exceção |
| ▣ Importância | ▣ Pós-condições |
| ▣ Sumário | ▣ Regras do Negócio |
| ▣ Ator Primário | ▣ Histórico |
| ▣ Atores Secundários | ▣ Notas de Implementação |
| ▣ Pré-condições | |

Boas práticas na documentação de casos de uso: Comece o nome do caso de uso com um verbo no infinitivo (para indicar um processo ou ação). Tente descrever os passos do caso sempre na forma sujeito + predicado. Ou seja, deixe explícito quem é o agente da ação. Não descreva como o sistema realiza internamente um passo de um caso de uso. Tente dar nomes a casos de uso seguindo perspectiva do ator primário. Foque no objetivo desse ator. Exemplos: Registrar Pedido, Abrir Ordem de Produção, Manter Referência, Alugar Filme, etc.

Ator primário: Troca informações com o sistema

Ator secundário: Executa a operação.

Regras do Negócio: São políticas, condições ou restrições que devem ser consideradas na execução dos processos de uma organização. Descrevem a maneira pela qual a organização funciona. Estas regras são identificadas e documentadas no chamado modelo de regras do negócio (MRN). A descrição do modelo de regras do negócio pode ser feita utilizando-se texto informal, ou através de alguma forma de estruturação. Regras do negócio normalmente influenciam o comportamento de determinados casos de uso.

MCU no processo de desenvolvimento: 1) Risco alto e prioridade alta 2) Risco alto e prioridade baixa 3) Risco baixo e prioridade alta 4) Risco baixo e prioridade baixa.

Preâmbulo:

- **ID:** identificador do caso de uso
- **Caso de uso:** nome do caso de uso
- **Atores Primários e Secundários:** atores envolvidos
- **Propósito:** frase que resume o caso de uso.
- **Visão Geral:** descrição curta da funcionalidade representada pelo caso de uso.
- **Pré-condições:** quais as pré-condições para que esse caso de uso possa ser realizado.
- **Pós-condições:** o que será obtido após a realização deste caso de uso.
- **Referências cruzadas:** referências a requisitos e outros casos de uso.

- ❑ **Fluxo Básico ou Principal:** tabela que descreve a interação principal do(s) ator(es) com o sistema para o caso de uso.

Ação do Ator	Resposta do Sistema
1. ...	2. ...
3. ...	
4.

- ❑ **Fluxos Alternativos:** descrição de possíveis interações caso o fluxo principal não seja possível.

- ❑ **Relembrando:** os relacionamentos podem ser de três tipos:

- 1) **Entre atores e casos de uso:**
 - ❑ Associação
- 2) **Entre casos de uso:**
 - ❑ Extend
 - ❑ Include
 - ❑ Generalização
- 3) **Entre atores:**
 - ❑ Generalização

Exemplo 1

- ❑ **Fluxos Alternativos (o que pode dar errado?)**

- **Passos 3, 5:** O administrador deseja cancelar a operação, acionando a opção de retornar ao navegador; sistema exibe página inicial do site.
- **Passo 5:** O administrador insere senha incorreta; sistema exibe mensagem de senha inválida, retornando à opção de efetuar o login de administrador.
- **Passo 7, 9, 10:** O administrador deseja cancelar a operação, acionando a opção de retornar do navegador; sistema exibe página de administração do site.
- **Passo 9:** O administrador insere dados incorretos (dados com formatos inválidos); sistema exibe mensagem de erro; o formulário é limpo e reexibido.
- **Passo 10:** A categoria já está cadastrada no sistema; sistema exibe mensagem de aviso, retornando à página de administração.