

ENGENHARIA DE SOFTWARE



Engenharia de Requisitos
Prof. Giovani Fonseca Ravagnani Disperati
IFSP – Câmpus Guarulhos

CONTEÚDO PROGRAMÁTICO



- Introdução à Engenharia de Requisitos
- Requisitos funcionais e não funcionais
- Processos de engenharia de requisitos
- Descoberta de requisitos
- Especificação de requisitos
- Validação de requisitos
- Mudança de requisitos

INTRODUÇÃO À ENGENHARIA DE REQUISITOS

Engenharia de Requisitos



- O processo de estabelecer os serviços que um cliente requer de um sistema e as restrições sob as quais ele opera e é desenvolvido.
- Os requisitos do sistema são as descrições dos serviços do sistema e restrições que são gerados durante o processo de engenharia de requisitos.

Engenharia de Requisitos



- O que é um requisito?
 - Uma requisito pode variar de uma declaração abstrata de alto nível de um serviço ou de uma restrição do sistema a uma especificação funcional matemática detalhada.

Engenharia de Requisitos



- Isso é inevitável, pois os requisitos podem ter uma função dupla
 - Podem ser a base para uma licitação de um contrato - portanto, devem estar abertos a interpretação;
 - Podem ser a base do próprio contrato - portanto, devem ser definidos em detalhes;
- Ambas as declarações podem ser chamadas de requisitos.

Engenharia de Requisitos



- Sommerville afirma que há dois tipos de requisitos: requisitos do usuário e requisitos do sistema.
- Requisitos do usuário: declarações em linguagem natural mais diagramas dos serviços que o sistema fornece e suas restrições operacionais. Escrito para clientes.
- Requisitos de sistema: um documento estruturado que estabelece descrições detalhadas das funções, serviços e restrições operacionais do sistema. Define o que deve ser implementado para fazer parte de um contrato entre o cliente e o contratado.

Engenharia de Requisitos



User requirements definition

- 1.** The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

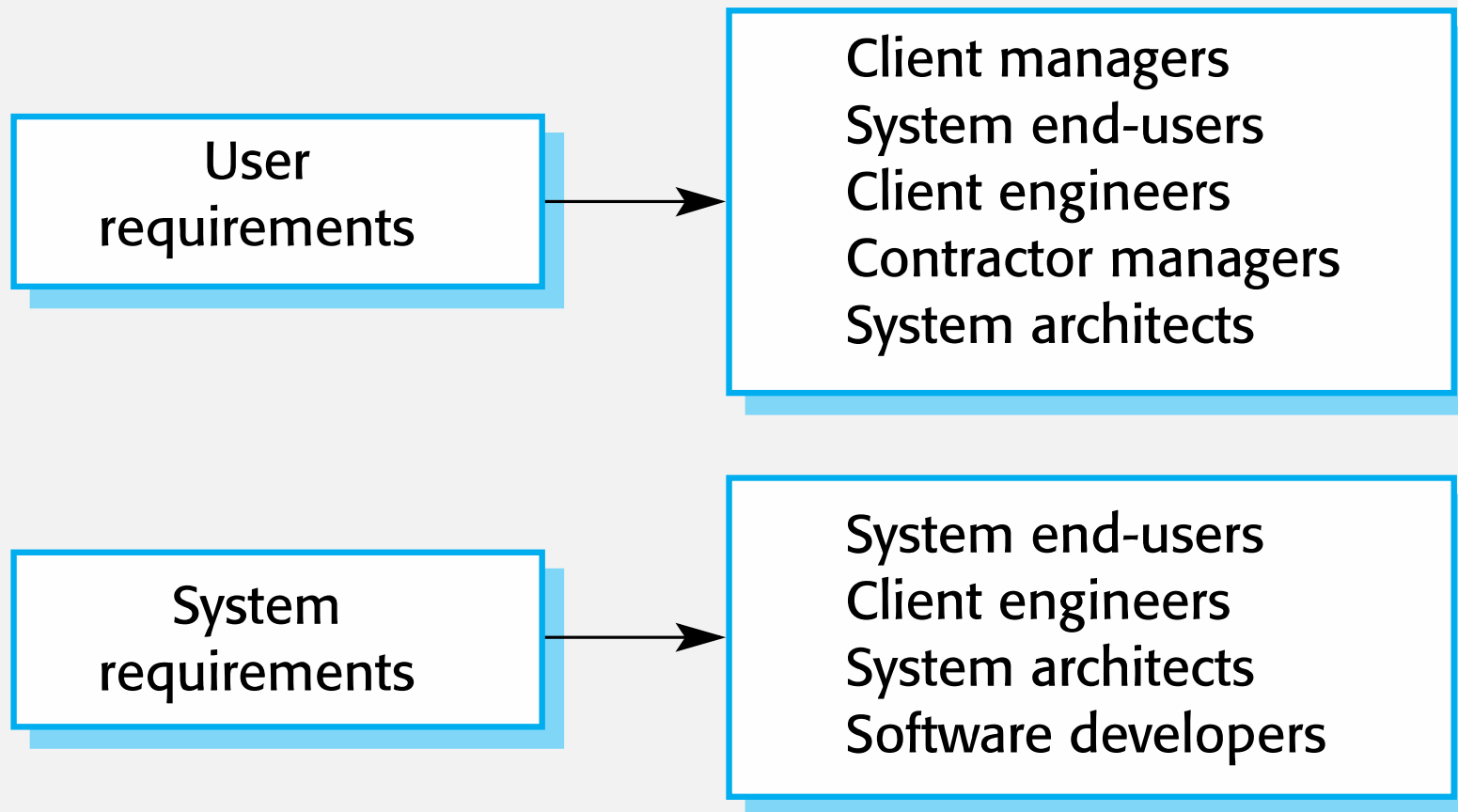
System requirements specification

- 1.1** On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- 1.2** The system shall generate the report for printing after 17.30 on the last working day of the month.
- 1.3** A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
- 1.4** If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
- 1.5** Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Engenharia de Requisitos



- Os requisitos são lidos por diferentes tipos de usuários.



Engenharia de Requisitos



- Stakeholder é qualquer pessoa ou organização que seja afetada pelo sistema de alguma forma e, portanto, tenha um interesse legítimo
- Tipos de partes interessadas
 - Usuários finais
 - Gerentes de sistema
 - Proprietários do sistema
 - Partes interessadas externas

Engenharia de Requisitos



- No sistema de Mentcare, possíveis stakeholders seriam:
 - Pacientes cujas informações são registradas no sistema.
 - Médicos responsáveis pela avaliação e tratamento dos pacientes.
 - Enfermeiros que coordenam as consultas com os médicos e administram alguns tratamentos.
 - Recepcionistas médicas que gerenciam as consultas dos pacientes.
 - Equipe de TI responsável pela instalação e manutenção do sistema.

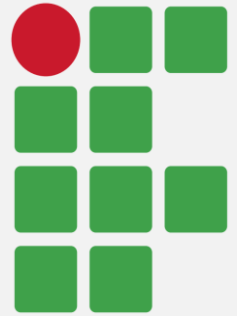
Engenharia de Requisitos



- Muitos métodos ágeis argumentam que produzir requisitos de sistema detalhados é uma perda de tempo, pois os requisitos mudam rapidamente
- O documento de requisitos está, portanto, sempre desatualizado
- Métodos ágeis geralmente usam engenharia de requisitos incrementais e podem expressar requisitos como "histórias de usuário" (discutidas no Capítulo 3)

Engenharia de Requisitos

- Isso é prático para sistemas de negócios, mas problemático para sistemas que requerem análise pré-entrega (por exemplo, sistemas críticos) ou sistemas desenvolvidos por várias equipes



INTRODUÇÃO AOS REQUISITOS FUNCIONAIS E NÃO- FUNCIONAIS

Requisitos funcionais e não-funcionais



- Requisitos funcionais são declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em situações específicas.

Requisitos funcionais e não-funcionais



- Requisitos funcionais descrevem a funcionalidade ou os serviços do sistema.
 - Depende do tipo de software, dos usuários esperados e do tipo de sistema em que o software é usado.
- Os requisitos funcionais do usuário podem ser declarações de alto nível do que o sistema deve fazer.
- Os requisitos do sistema devem descrever os serviços do sistema em detalhes.

Requisitos funcionais e não-funcionais



- Exemplo de requisitos funcionais no sistema de Mentcare:
 - O usuário poderá pesquisar as listas de consultas de todas as clínicas.
 - O sistema deve gerar a cada dia, para cada clínica, uma lista de pacientes que devem comparecer às consultas naquele dia.
 - Cada membro da equipe que usa o sistema deve ser identificado exclusivamente por seu número de funcionário de 8 dígitos.

Requisitos funcionais e não-funcionais



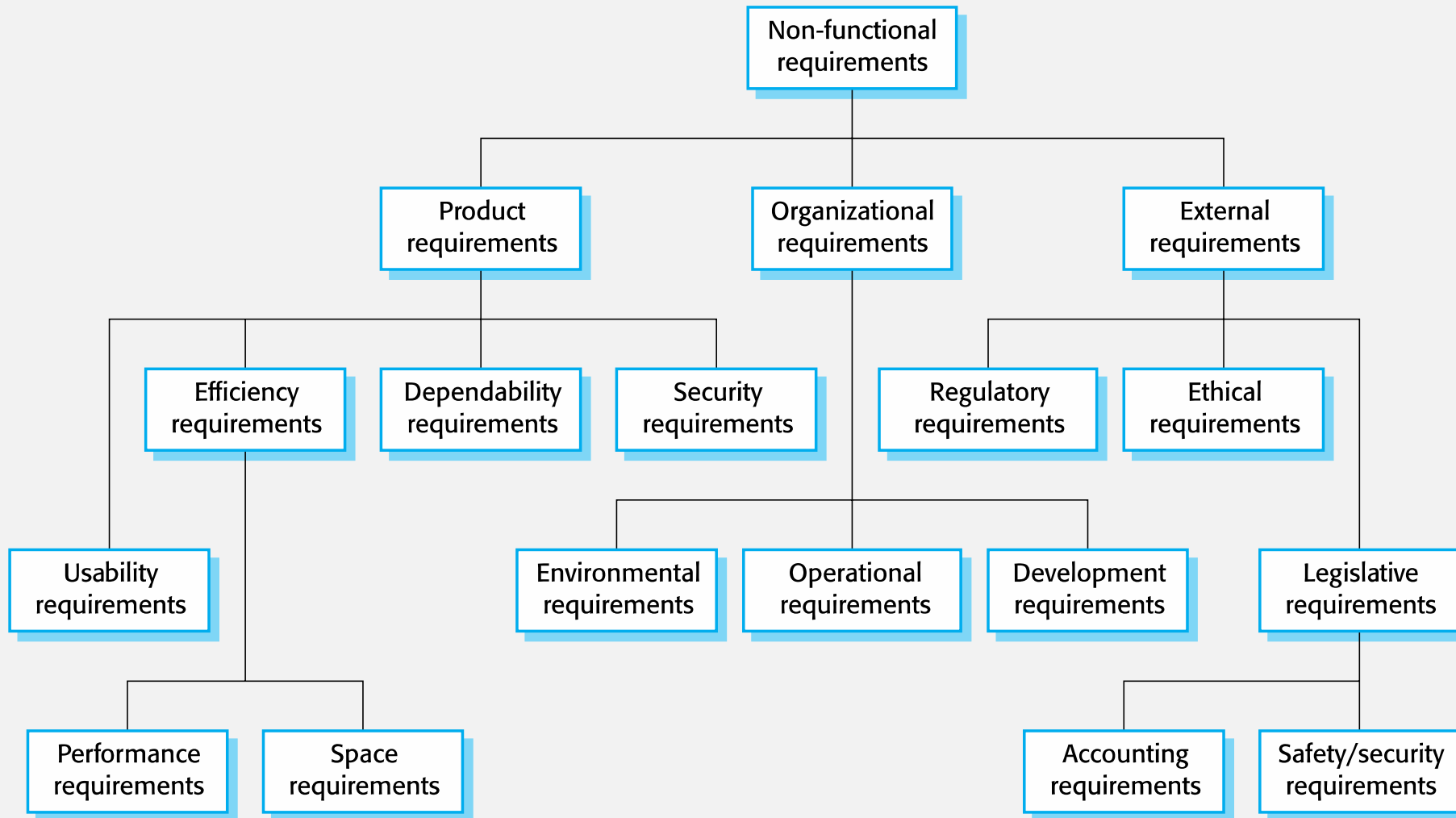
- Requisitos não funcionais são restrições nos serviços ou funções oferecidos pelo sistema, como restrições de tempo, restrições no processo de desenvolvimento, padrões, etc.
- Geralmente se aplicam ao sistema como um todo, em vez de recursos ou serviços individuais.

Requisitos não-funcionais

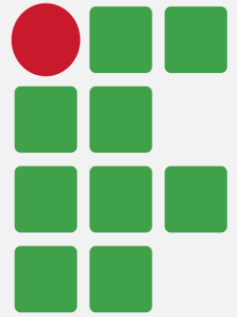


- Definem as propriedades e restrições do sistema, por exemplo: confiabilidade, tempo de resposta e requisitos de armazenamento.
- Os requisitos do processo também podem ser especificados exigindo um determinado IDE, linguagem de programação ou método de desenvolvimento.
- Requisitos não funcionais podem ser mais críticos do que requisitos funcionais. Se estes não forem atendidos, o sistema pode ser inútil.

Requisitos não-funcionais



Implementação de requisitos não-funcionais



- Os requisitos não funcionais podem afetar a arquitetura geral de um sistema em vez dos componentes individuais.
- Por exemplo, para garantir que os requisitos de desempenho sejam atendidos, pode ser necessário organizar o sistema para minimizar a comunicação entre os componentes.

Implementação de requisitos não-funcionais



- Um único requisito não funcional, como um requisito de segurança, pode gerar uma série de requisitos funcionais relacionados que definem os serviços do sistema que são necessários.
- Também pode gerar requisitos que restringem os requisitos existentes.

Classificação de requisitos não-funcionais



- Requisitos do produto
 - Requisitos que especificam que o produto entregue deve se comportar de uma maneira particular, por ex. velocidade de execução, confiabilidade, etc.
- Requisitos organizacionais
 - Requisitos que são consequência de políticas e procedimentos organizacionais, por exemplo padrões de processo usados, requisitos de implementação, etc.
- Requisitos externos
 - Requisitos que surgem de fatores externos ao sistema e seu processo de desenvolvimento, por exemplo requisitos de interoperabilidade, requisitos legislativos, etc.

Requisitos não-funcionais no sistema de Mentcare



Product requirement

The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

Organizational requirement

Users of the Mentcare system shall authenticate themselves using their health authority identity card.

External requirement

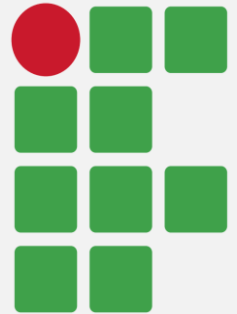
The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Metas e requisitos



- Os requisitos não funcionais podem ser muito difíceis de definir com precisão e os requisitos imprecisos podem ser difíceis de verificar.
- Objetivo
 - Uma intenção geral do usuário, como facilidade de uso.
- Requisito não funcional verificável
 - Uma declaração usando alguma medida que pode ser testada objetivamente.
- As metas são úteis para os desenvolvedores, pois transmitem as intenções dos usuários do sistema.

Métricas para especificar requisitos não-funcionais



Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

DESAFIOS NO LEVANTAMENTO DE REQUISITOS

Imprecisão de requisitos



- Os problemas surgem quando os requisitos funcionais não são definidos com precisão.
- Requisitos ambíguos podem ser interpretados de maneiras diferentes por desenvolvedores e usuários.

Imprecisão de requisitos



- Considere o termo "pesquisar" no requisito 1
- O usuário poderá pesquisar as listas de consultas de todas as clínicas.
- O sistema deve gerar a cada dia, para cada clínica, uma lista de pacientes que devem comparecer às consultas naquele dia.
- Cada membro da equipe que usa o sistema deve ser identificado exclusivamente por seu número de funcionário de 8 dígitos.

Completude e consistência de requisitos



- Em princípio, os requisitos devem ser completos e consistentes.
- Completo
 - Eles devem incluir descrições de tudo que for necessário.
- Consistente
 - Não deve haver conflitos ou contradições nas descrições dos recursos do sistema.

Completude e consistência de requisitos



- Na prática, devido à complexidade do sistema e do ambiente, é impossível produzir um documento de requisitos completo e consistente.

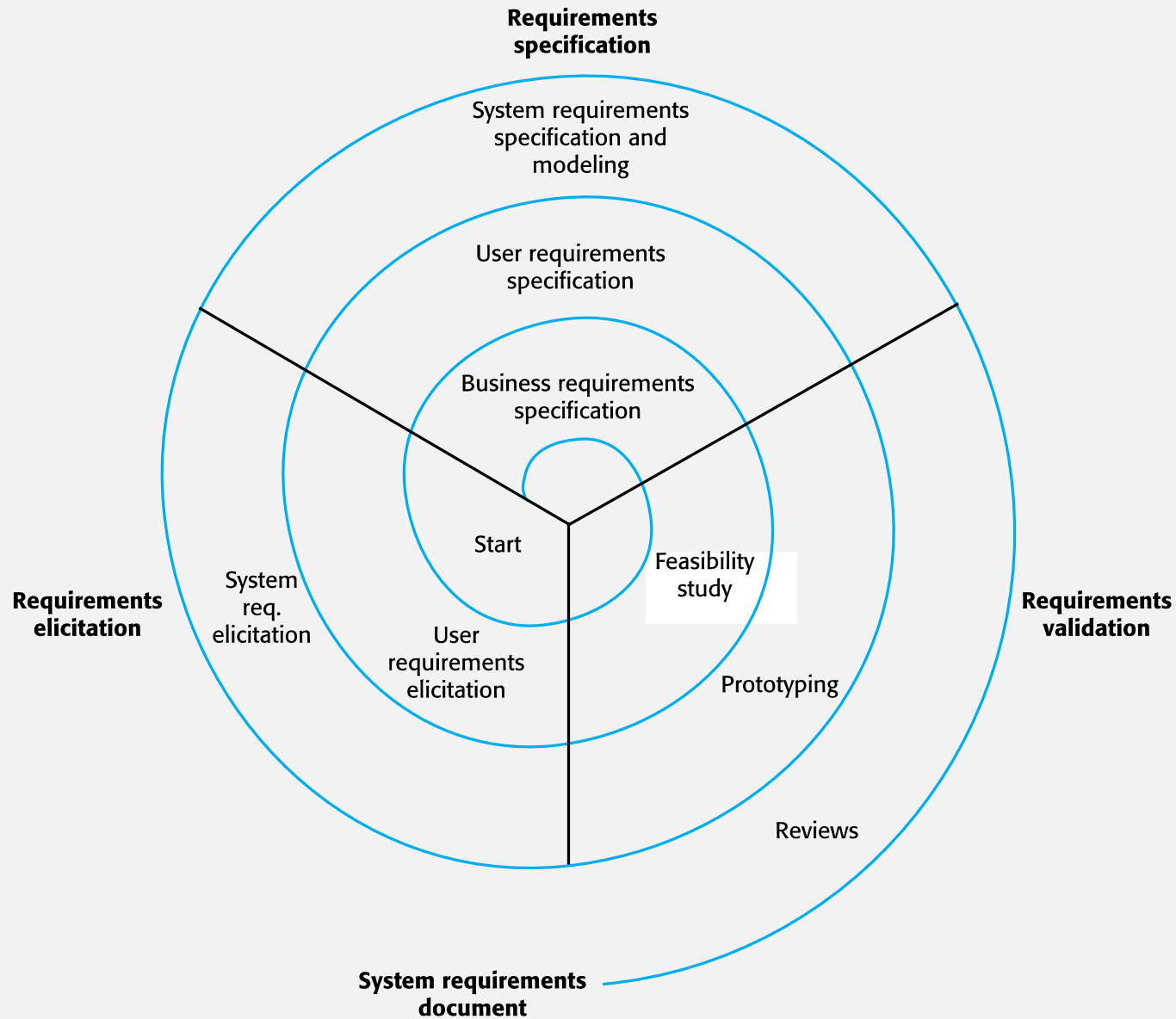
PROCESSOS DE ENGENHARIA DE REQUISITOS

Processos de engenharia de requisitos



- Os processos usados para ER variam amplamente, dependendo do domínio do aplicativo, das pessoas envolvidas e da organização que desenvolve os requisitos.
- No entanto, há uma série de atividades genéricas comuns a todos os processos
 - Levantamento de requisitos;
 - Análise de requisitos;
 - Validação de requisitos;
 - Gerenciamento de requisitos.
- Na prática, ER é uma atividade iterativa em que esses processos são intercalados.

Processos de engenharia de requisitos



LEVANTAMIENTO DE REQUISITOS

Levantamento de requisitos



- Às vezes chamado de elicitación de requisitos ou descoberta de requisitos.
- Envolve a equipe técnica que trabalha com os clientes para descobrir sobre o domínio do aplicativo, os serviços que o sistema deve fornecer e as restrições operacionais do sistema.
- Pode envolver usuários finais, gerentes, engenheiros envolvidos na manutenção, especialistas de domínio, sindicatos, etc. Esses são chamados de partes interessadas.

Levantamento de requisitos



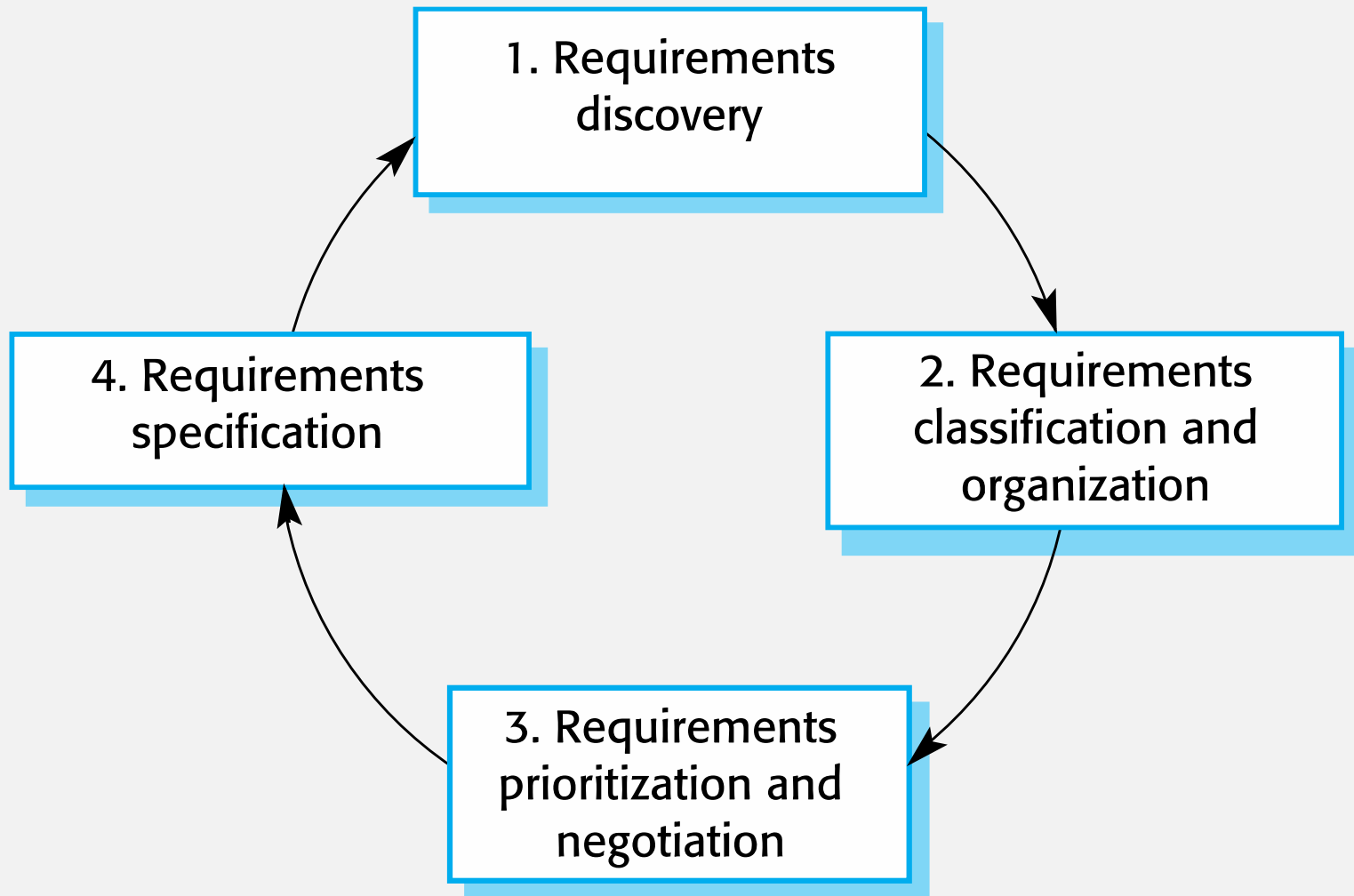
- Os engenheiros de software trabalham com uma série de partes interessadas no sistema para descobrir mais sobre o domínio do aplicativo, os serviços que o sistema deve fornecer, o desempenho do sistema necessário, restrições de hardware, outros sistemas, etc.
- As fases incluem:
 - Descoberta de requisitos,
 - Classificação e organização de requisitos,
 - Priorização e negociação de requisitos,
 - Especificação de requisitos.

Problemas de levantamento de requisitos



- As partes interessadas não sabem o que realmente querem.
- As partes interessadas expressam os requisitos em seus próprios termos.
- Diferentes partes interessadas podem ter requisitos conflitantes.
- Fatores organizacionais e políticos podem influenciar os requisitos do sistema.
- Os requisitos mudam durante o processo de análise. Novos stakeholders podem surgir e o ambiente de negócios pode mudar.

Atividades do processo



Atividades do processo



- Descoberta de requisitos
 - Interação com as partes interessadas para descobrir seus requisitos. Os requisitos de domínio também são descobertos neste estágio.
- Classificação e organização de requisitos
 - Agrupa requisitos relacionados e os organiza em clusters coerentes.
- Priorização e negociação
 - Prioriza requisitos e resolve conflitos de requisitos.
- Especificação de requisitos
 - Os requisitos são documentados e inseridos na próxima rodada da espiral.

DESCOBERTA DE REQUISITOS

Descoberta de requisitos



- O processo de coleta de informações sobre os sistemas necessários e existentes e destilação dos requisitos do usuário e do sistema a partir dessas informações
- A interação é com as partes interessadas do sistema, desde gerentes a reguladores externos
- Os sistemas normalmente possuem uma variedade de partes interessadas

ENTREVISTAS

Entrevistas



- Entrevistas formais ou informais com as partes interessadas fazem parte da maioria dos processos de ER.
- Tipos de entrevista:
- Entrevistas fechadas com base em uma lista pré-determinada de perguntas
- Entrevistas abertas onde várias questões são exploradas com as partes interessadas.

Entrevistas



- Entrevista eficaz
- Tenha a mente aberta, evite ideias pré-concebidas sobre os requisitos e esteja disposto a ouvir as partes interessadas.
- Peça ao entrevistado para iniciar as discussões usando uma questão de trampolim, uma proposta de requisitos ou trabalhando junto em um sistema de protótipo.

Entrevistas na prática



- Normalmente uma mistura de entrevistas fechadas e abertas
- As entrevistas são boas para obter uma compreensão geral do que as partes interessadas fazem e como podem interagir com o sistema
- Os entrevistadores precisam ter a mente aberta, sem ideias pré-concebidas do que o sistema deve fazer
- Você precisa solicitar que o usuário fale sobre o sistema, sugerindo requisitos em vez de simplesmente perguntar o que eles desejam

Problemas com entrevistas



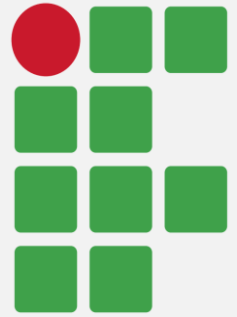
- Os especialistas nas aplicações podem usar uma linguagem para descrever seu trabalho que não é fácil para o engenheiro de requisitos entender.
- Entrevistas não são boas para entender os requisitos de domínio
 - Os engenheiros de requisitos podem não entender a terminologia de domínio específica;
 - Alguns conhecimentos de domínio são tão familiares que as pessoas acham difícil articulá-los ou pensam que não vale a pena articular.

ETNOGRAFIA

Etnografia



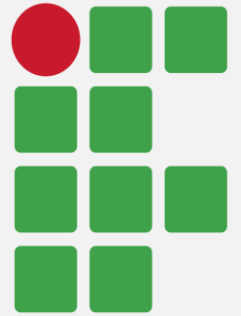
- Um cientista social passa um tempo considerável observando e analisando como as pessoas realmente trabalham
- As pessoas não precisam explicar ou articular seu trabalho
- Fatores sociais e organizacionais de importância podem ser observados
- Estudos etnográficos têm mostrado que o trabalho é geralmente mais rico e complexo do que o sugerido por modelos de sistema simples



Escopo da etnografia

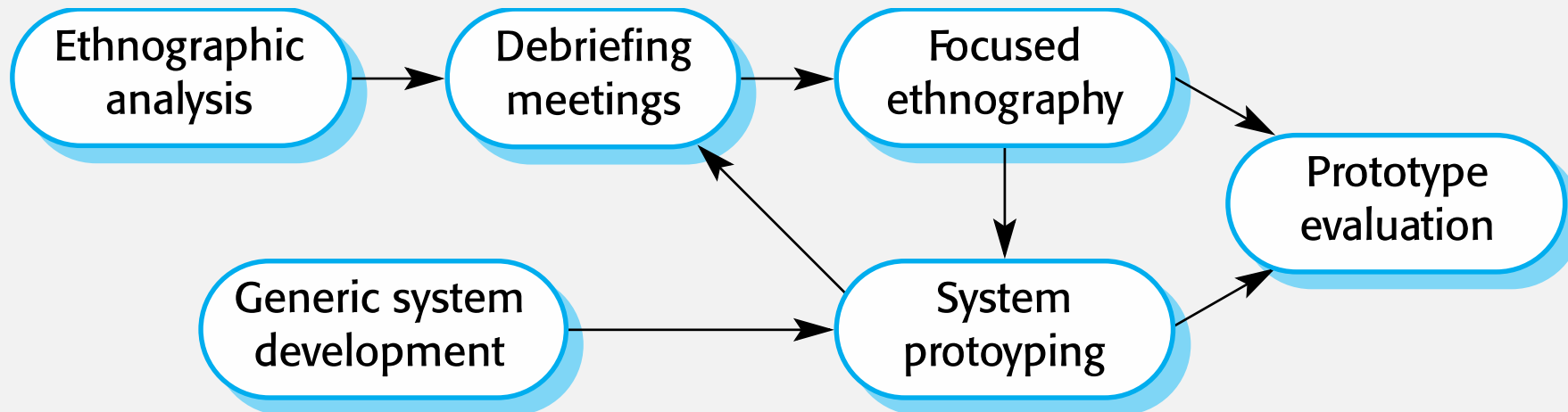
- Requisitos que são derivados da maneira como as pessoas realmente trabalham, e não da maneira como as definições de processo sugerem que elas devem funcionar.
- Requisitos que são derivados da cooperação e consciência das atividades de outras pessoas.
 - A consciência do que as outras pessoas estão fazendo leva a mudanças na maneira como fazemos as coisas.
- A etnografia é eficaz para compreender os processos existentes, mas não pode identificar novos recursos que devem ser adicionados a um sistema.

Etnografia focada



- Combina etnografia com prototipagem
- O desenvolvimento de protótipos resulta em questões não respondidas que focam a análise etnográfica
- O problema da etnografia é que ela estuda práticas existentes que podem ter alguma base histórica que não é mais relevante

Etnografia e prototipagem



HISTÓRIAS E CENÁRIOS

Histórias e cenários

- Jack é professor primário em Ullapool (uma vila no norte da Escócia). Ele decidiu que um projeto de classe deveria ser focado em torno da indústria pesqueira na área, olhando para a história, desenvolvimento e impacto econômico da pesca. Como parte disso, os alunos são convidados a reunir e compartilhar reminiscências de parentes, usar arquivos de jornais e coletar fotos antigas relacionadas à pesca e às comunidades pesqueiras da área. Os alunos usam um wiki iLearn para reunir histórias de pesca e SCRAN (um site de recursos de história) para acessar arquivos de jornais e fotografias. No entanto, Jack também precisa de um site de compartilhamento de fotos, pois ele deseja que os alunos tirem e comentem as fotos uns dos outros e carreguem scans de fotos antigas que possam ter em suas famílias.



Histórias e cenários

- Jack envia um e-mail para um grupo de professores de escola primária, do qual ele é membro, para ver se alguém pode recomendar um sistema apropriado. Dois professores respondem e ambos sugerem que ele use o KidsTakePics, um site de compartilhamento de fotos que permite aos professores verificar e moderar o conteúdo. Como o KidsTakePics não está integrado ao serviço de autenticação iLearn, ele configura um professor e uma conta para a turma. Ele usa o serviço de configuração iLearn para adicionar o KidsTakePics aos serviços vistos pelos alunos em sua turma para que, quando eles se conectem, possam usar o sistema imediatamente para enviar fotos de seus dispositivos móveis e computadores da turma.



Histórias e cenários



- Uma forma estruturada de história do usuário
- Os cenários devem incluir
 - Uma descrição da situação inicial;
 - Uma descrição do fluxo normal de eventos;
 - Uma descrição do que pode dar errado;
 - Informações sobre outras atividades simultâneas;
 - Uma descrição do estado quando o cenário termina.

Histórias e cenários: upload de fotos (ILearn)



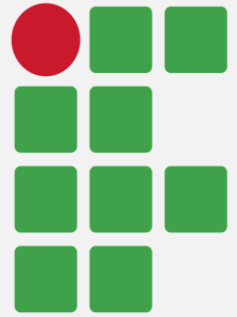
- Pressuposto inicial: um usuário ou grupo de usuários tem uma ou mais fotografias digitais a serem carregadas no site de compartilhamento de imagens. Eles são salvos em um tablet ou laptop. Eles se conectaram com sucesso ao KidsTakePics.
- Normal: o usuário escolhe fazer upload de fotos e é solicitado a selecionar as fotos a serem carregadas em seu computador e a selecionar o nome do projeto no qual as fotos serão armazenadas. Eles também devem ter a opção de inserir palavras-chave que devem ser associadas a cada foto enviada. As fotos carregadas são nomeadas criando uma conjunção do nome do usuário com o nome do arquivo da foto no computador local.
- Após a conclusão do upload, o sistema automaticamente envia um e-mail para o moderador do projeto solicitando a verificação do novo conteúdo e gera uma mensagem na tela para o usuário informando que isso foi feito.

Histórias e cenários: upload de fotos (ILearn)



- O que pode dar errado:
- Nenhum moderador está associado ao projeto selecionado. Um e-mail é gerado automaticamente para o administrador da escola, pedindo-lhes para nomear um moderador do projeto. Os usuários devem ser informados de que pode haver um atraso em tornar suas fotos visíveis.
- Fotos com o mesmo nome já foram enviadas pelo mesmo usuário. O usuário deve ser questionado se deseja reenviar as fotos com o mesmo nome, renomear as fotos ou cancelar o envio. Se eles escolherem reenviar as fotos, as originais serão substituídas. Se eles escolherem renomear as fotos, um novo nome é gerado automaticamente adicionando um número ao nome do arquivo existente.

Histórias e cenários: upload de fotos (ILearn)



- Outras atividades: O moderador pode estar conectado ao sistema e pode aprovar as fotos à medida que são carregadas.
- Estado do sistema na conclusão: o usuário está conectado. As fotos selecionadas foram enviadas e atribuídas ao status "aguardando moderação". As fotos são visíveis para o moderador e para o usuário que as carregou.

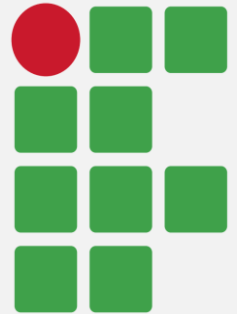
ESPECIFICAÇÃO DE REQUISITOS

Especificação de Requisitos



- O processo de escrever os requisitos do usuário e do sistema em um documento de requisitos.
- Os requisitos do usuário devem ser compreensíveis pelos usuários finais e clientes que não possuem formação técnica.
- Os requisitos do sistema são requisitos mais detalhados e podem incluir mais informações técnicas.
- Os requisitos podem fazer parte de um contrato para o desenvolvimento do sistema
 - Portanto, é importante que sejam o mais completo possível.

Formas de escrever especificações de requisitos



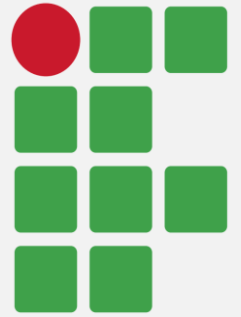
Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

Requisitos e Design



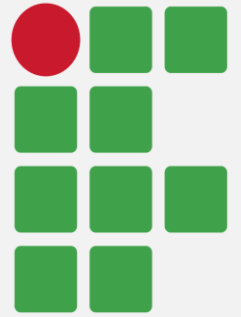
- Em princípio, os requisitos devem definir o que o sistema deve fazer e o projeto deve descrever como ele faz isso.
- Na prática, requisitos e design são inseparáveis
 - Uma arquitetura de sistema pode ser projetada para estruturar os requisitos;
 - O sistema pode interoperar com outros sistemas que geram requisitos de projeto;
 - O uso de uma arquitetura específica para satisfazer requisitos não funcionais pode ser um requisito de domínio.
 - Isso pode ser consequência de uma exigência regulatória.

ESPECIFICAÇÃO DE REQUISITOS EM LINGUAGEM NATURAL



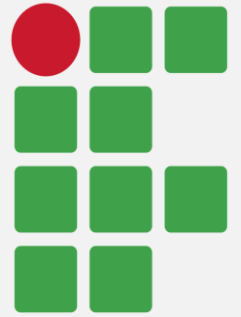
Especificação em linguagem natural

- Os requisitos são escritos como frases em linguagem natural complementadas por diagramas e tabelas.
- Usado para escrever requisitos porque é expressivo, intuitivo e universal. Isso significa que os requisitos podem ser compreendidos por usuários e clientes.



Guias para escrever requisitos

- Invente um formato padrão e use-o para todos os requisitos.
- Use a linguagem de maneira consistente. Use “deve” para requisitos obrigatórios, “pode” para requisitos desejáveis.
- Use o realce de texto para identificar as partes principais do requisito.
- Evite o uso de jargões de informática.
- Inclua uma explicação (justificativa) de por que um requisito é necessário.



Problemas com linguagem natural

- Falta de clareza
 - A precisão é difícil sem tornar o documento difícil de ler.
- Confusão de requisitos
 - Requisitos funcionais e não funcionais tendem a se misturar.
- Amalgamação de requisitos
 - Vários requisitos diferentes podem ser expressos juntos.

Exemplo de requisitos em um sistema de bomba de insulina



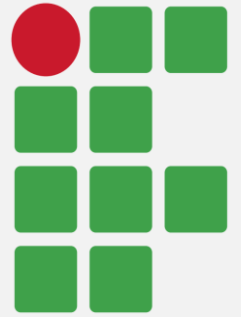
- 3.2 The system shall measure the blood sugar and deliver insulin, if required, every 10 minutes. (Changes in blood sugar are relatively slow so more frequent measurement is unnecessary; less frequent measurement could lead to unnecessarily high sugar levels.)
- 3.6 The system shall run a self-test routine every minute with the conditions to be tested and the associated actions defined in Table 1. (A self-test routine can discover hardware and software problems and alert the user to the fact the normal operation may be impossible.)

ESPECIFICAÇÃO DE REQUISITOS POR MEIO DE ESPECIFICAÇÕES ESTRUTURADAS

Especificações estruturadas



- Uma abordagem para escrever requisitos em que a liberdade do redator de requisitos é limitada e os requisitos são escritos de uma forma padrão.
- Isso funciona bem para alguns tipos de requisitos, por exemplo, requisitos para sistema de controle embarcados, mas às vezes é muito rígido para escrever requisitos de sistema de negócios.



Especificações estruturadas

- Definição da função ou entidade.
- Descrição das entradas e de onde vêm.
- Descrição das saídas e para onde vão.
- Informações sobre as informações necessárias para o cálculo e outras entidades usadas.
- Descrição da ação a ser executada.
- Pré e pós-condições (se apropriado).
- Os efeitos colaterais (se houver) da função.

Exemplo no sistema de bomba de insulina



Insulin Pump/Control Software/SRS/3.3.2

Function Compute insulin dose: safe sugar level.

Description

Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

Inputs Current sugar reading (r2); the previous two readings (r0 and r1).

Source Current sugar reading from sensor. Other readings from memory.

Outputs CompDose—the dose in insulin to be delivered.

Destination Main control loop.

Exemplo no sistema de bomba de insulina



Action

CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

Requirements

Two previous readings so that the rate of change of sugar level can be computed.

Pre-condition

The insulin reservoir contains at least the maximum allowed single dose of insulin.

Post-condition r_0 is replaced by r_1 then r_1 is replaced by r_2 .

Side effects None.

ESPECIFICAÇÃO DE REQUISITOS POR MEIO DE ESPECIFICAÇÕES TABULARES

Especificações tabulares



- Usado para complementar a linguagem natural.
- Particularmente útil quando você precisa definir uma série de possíveis cursos de ação alternativos.
- Por exemplo, o sistema de bomba de insulina baseia seus cálculos na taxa de variação do nível de açúcar no sangue e a especificação tabular explica como calcular a necessidade de insulina para diferentes cenários.

Exemplo de especificação tabular



Condition	Action
Sugar level falling ($r2 < r1$)	CompDose = 0
Sugar level stable ($r2 = r1$)	CompDose = 0
Sugar level increasing and rate of increase decreasing ($(r2 - r1) < (r1 - r0)$)	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ($(r2 - r1) \geq (r1 - r0)$)	CompDose = round $((r2 - r1)/4)$ If rounded result = 0 then CompDose = MinimumDose

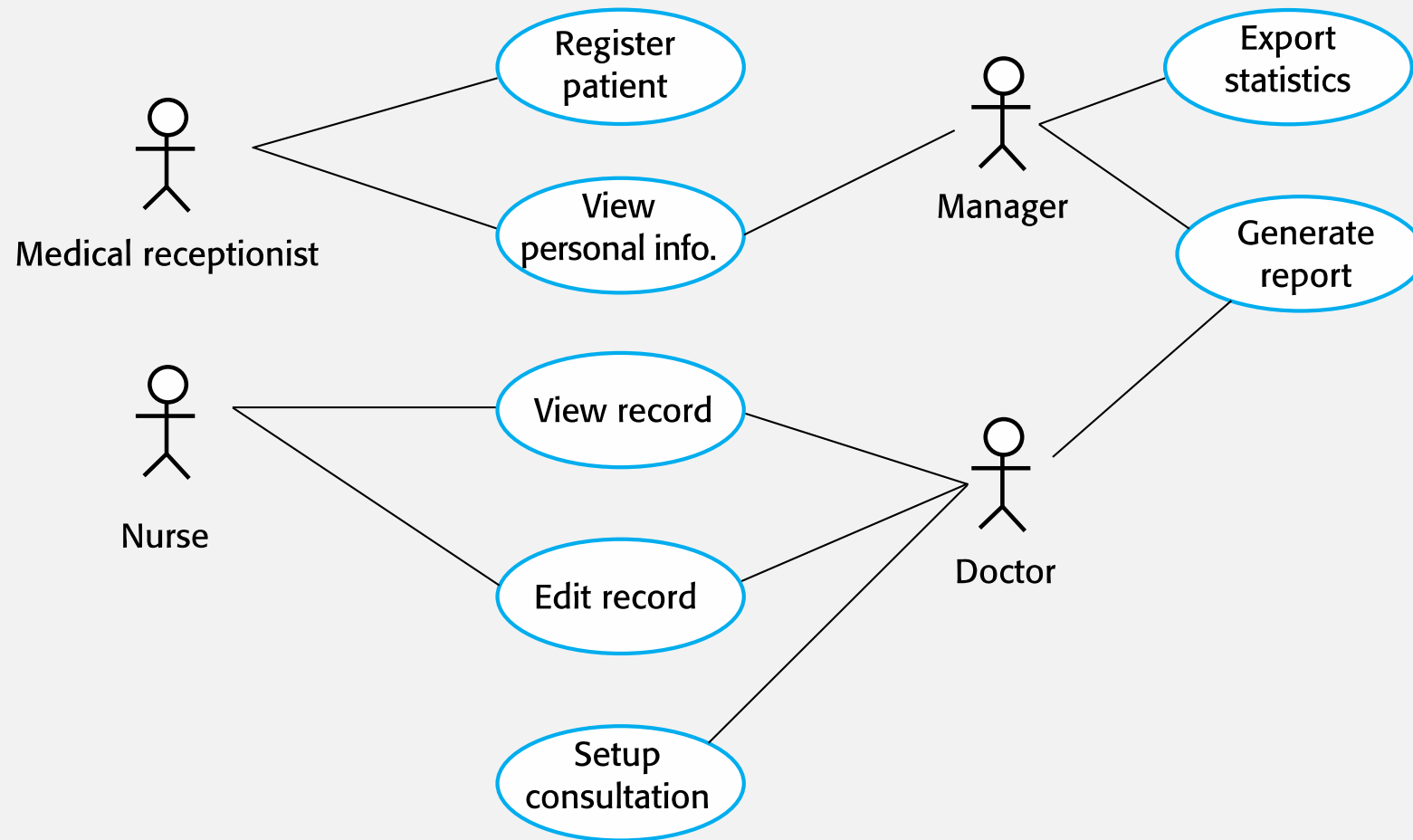
ESPECIFICAÇÃO DE REQUISITOS POR MEIO DE CASOS DE USO

Casos de uso

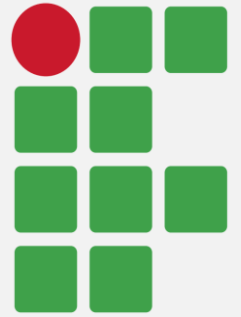


- Os casos de uso são um tipo de cenário incluído na UML.
- Os casos de uso identificam os atores em uma interação e descrevem a própria interação.
- Um conjunto de casos de uso deve descrever todas as interações possíveis com o sistema.
- Modelo gráfico de alto nível complementado por uma descrição tabular mais detalhada (consulte o Capítulo 5).
- Os diagramas de sequência UML podem ser usados para adicionar detalhes aos casos de uso, mostrando a sequência de processamento de eventos no sistema.

Caso de uso para o sistema Mentcare



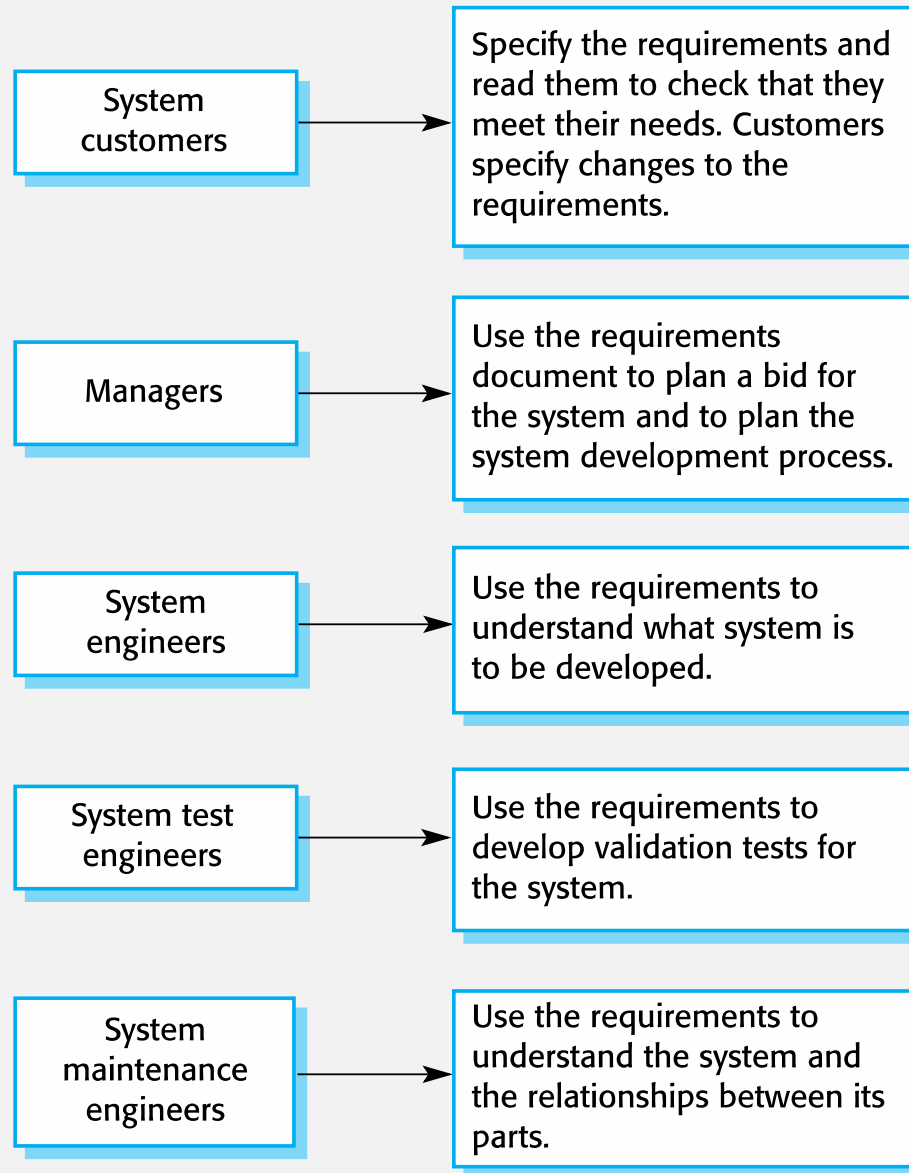
O DOCUMENTO DE REQUISITOS DE SOFTWARE



O documento de requisitos de software

- O documento de requisitos de software é a declaração oficial do que é exigido dos desenvolvedores de sistema.
- Deve incluir uma definição dos requisitos do usuário e uma especificação dos requisitos do sistema.
- NÃO é um documento de design. Na medida do possível, deve definir O QUE o sistema deve fazer ao invés de COMO deve fazê-lo.

0 documento de requisitos de software

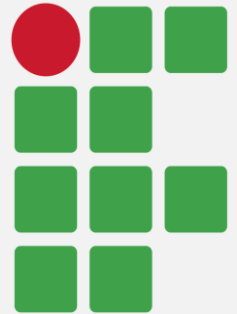


Variabilidade do documento de requisitos



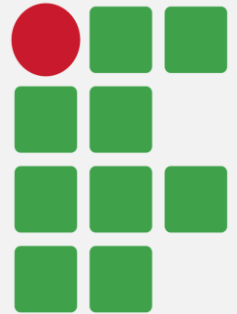
- As informações no documento de requisitos dependem do tipo de sistema e da abordagem de desenvolvimento usada.
- Os sistemas desenvolvidos de forma incremental terão, normalmente, menos detalhes no documento de requisitos.
- Padrões de documentos de requisitos foram projetados, por exemplo Padrão IEEE. Estes são principalmente aplicáveis aos requisitos para grandes projetos de engenharia de sistemas.

Estrutura de um documento de requisitos



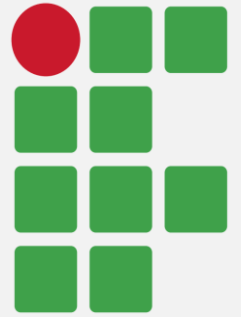
Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.

Estrutura de um documento de requisitos



Chapter	Description
System requirements specification	This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

VERIFICAÇÃO E VALIDAÇÃO DE REQUISITOS



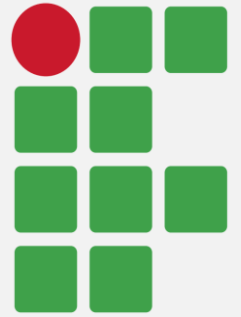
Validação de requisitos

- Preocupa-se em demonstrar que os requisitos definem o sistema que o cliente realmente deseja.
- Os custos de erro de requisitos são altos, então a validação é muito importante
 - Corrigir um erro de requisitos após a entrega pode custar até 100 vezes o custo de corrigir um erro de implementação.

Verificação de requisitos



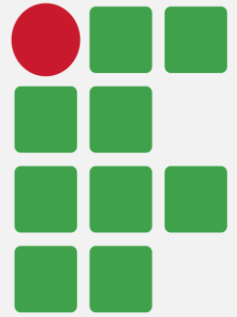
- Validade. O sistema fornece as funções que melhor atendem às necessidades do cliente?
- Consistência. Existem conflitos de requisitos?
- Completude. Todas as funções exigidas pelo cliente estão incluídas?
- Realismo. Os requisitos podem ser implementados de acordo com o orçamento e tecnologia disponíveis
- Verificabilidade. Os requisitos podem ser verificados?



Técnicas de validação de requisitos

- Revisões de requisitos
 - Análise manual sistemática dos requisitos.
- Prototipagem
 - Usando um modelo executável do sistema para verificar os requisitos. Abordado no Capítulo 2.
- Geração de casos de teste
 - Desenvolvimento de testes de requisitos para verificar a testabilidade.

Revisão de requisitos



- Revisões regulares devem ser realizadas enquanto a definição dos requisitos está sendo formulada.
- Tanto a equipe do cliente quanto a do contratado devem estar envolvidas nas revisões.
- As revisões podem ser formais (com documentos preenchidos) ou informais. Uma boa comunicação entre desenvolvedores, clientes e usuários pode resolver problemas em um estágio inicial.

Revisão de requisitos

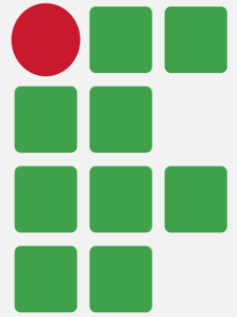


- Verificabilidade
 - O requisito é realisticamente testável?
- Compreensibilidade
 - O requisito foi devidamente compreendido?
- Rastreabilidade
 - A origem do requisito está claramente indicada?
- Adaptabilidade
 - O requisito pode ser alterado sem um grande impacto nos outros requisitos?

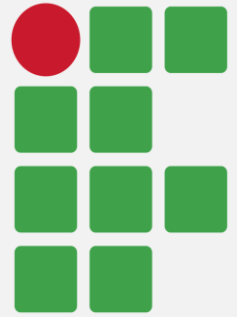
MUDANÇA DE REQUISITOS

Mudança de requisitos

- O ambiente comercial e técnico do sistema sempre muda após a instalação.
- Pode ser introduzido um novo hardware, pode ser necessário fazer a interface do sistema com outros sistemas, as prioridades do negócio podem mudar e podem ser introduzidas novas legislações e regulamentos que o sistema deve necessariamente cumprir.



Mudança de requisitos

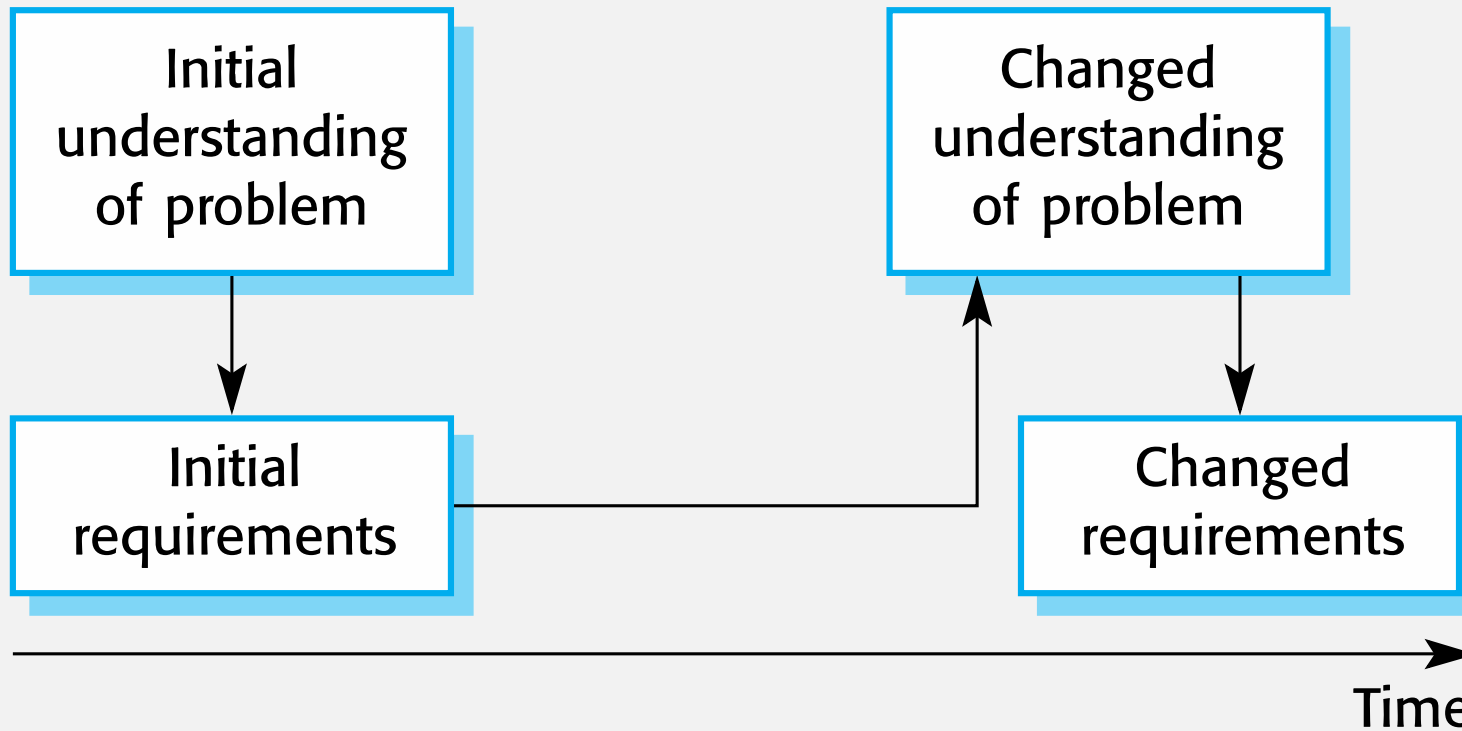


- As pessoas que pagam por um sistema e os usuários desse sistema raramente são as mesmas pessoas.
- Os clientes do sistema impõem requisitos devido a restrições organizacionais e orçamentárias. Eles podem entrar em conflito com os requisitos do usuário final e, após a entrega, novos recursos podem ter que ser adicionados para suporte ao usuário para que o sistema atenda aos seus objetivos.

Mudança de requisitos



- Os sistemas grandes geralmente têm uma comunidade de usuários diversificada, com muitos usuários com requisitos e prioridades diferentes que podem ser conflitantes ou contraditórios.
- Os requisitos finais do sistema são inevitavelmente um meio-termo entre eles e, com a experiência, muitas vezes se descobre que o equilíbrio do suporte fornecido a diferentes usuários deve ser alterado.



Gestão de requisitos



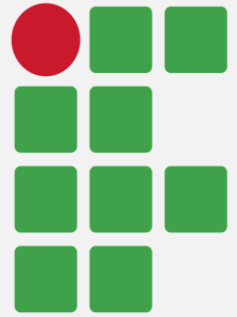
- O gerenciamento de requisitos é o processo de gerenciamento de mudanças de requisitos durante o processo de engenharia de requisitos e o desenvolvimento do sistema
- Novos requisitos surgem à medida que um sistema está sendo desenvolvido e depois de entrar em uso
- Você precisa acompanhar os requisitos individuais e manter links entre os requisitos dependentes para que possa avaliar o impacto das mudanças nos requisitos. Você precisa estabelecer um processo formal para fazer propostas de mudança e vinculá-las aos requisitos do sistema.

Planejamento de mudança de requisitos

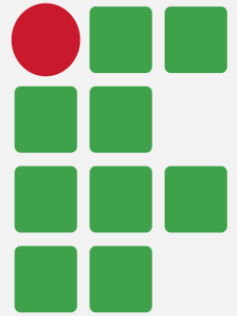


- Estabelece o nível de detalhes de gerenciamento de requisitos que é necessário.
- Decisões de gerenciamento de requisitos:
- Identificação de requisitos: cada requisito deve ser identificado exclusivamente para que possa ser cruzado com outros requisitos.
- Um processo de gerenciamento de mudanças: este é o conjunto de atividades que avalia o impacto e o custo das mudanças. Discuto esse processo com mais detalhes na seção a seguir.

Planejamento de mudança de requisitos



- Políticas de rastreabilidade: essas políticas definem os relacionamentos entre cada requisito e entre os requisitos e o design do sistema que devem ser registrados.
- Ferramentas de suporte: as ferramentas que podem ser usadas variam de sistemas especializados de gerenciamento de requisitos a planilhas e sistemas de banco de dados simples.



Gestão de mudança de requisitos

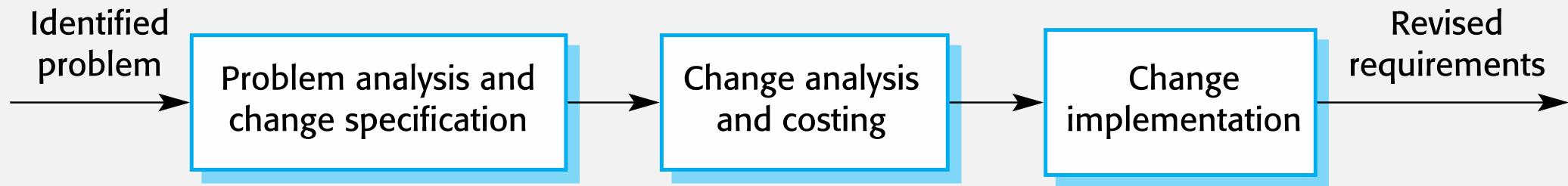
- Decidir se uma mudança de requisitos deve ser aceita
- Análise de problema e especificação de mudança
 - Nessa etapa, o problema ou a proposta de mudança é analisado para verificar se é válido. Essa análise é retornada ao solicitante de mudança, que pode responder com uma proposta de mudança de requisitos mais específica ou decidir retirar a solicitação.

Gestão de mudança de requisitos



- Análise de mudanças e custos
 - O efeito da mudança proposta é avaliado usando informações de rastreabilidade e conhecimento geral dos requisitos do sistema. Depois que essa análise for concluída, é tomada a decisão de prosseguir ou não com a alteração dos requisitos.
- Implementação de mudança
 - O documento de requisitos e, quando necessário, o design e a implementação do sistema são modificados. Idealmente, o documento deve ser organizado de forma que as mudanças possam ser facilmente implementadas

Gestão de mudança de requisitos



PONTOS CHAVE

Pontos chave



- Os requisitos para um sistema de software estabelecem o que o sistema deve fazer e definem as restrições à sua operação e implementação.
- Os requisitos funcionais são declarações dos serviços que o sistema deve fornecer ou são descrições de como alguns cálculos devem ser realizados.

Pontos chave



- Os requisitos não funcionais geralmente restringem o sistema que está sendo desenvolvido e o processo de desenvolvimento que está sendo usado.
- Frequentemente, estão relacionados às propriedades emergentes do sistema e, portanto, se aplicam ao sistema como um todo.

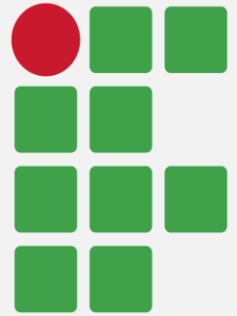
Pontos chave



- O processo de engenharia de requisitos é um processo iterativo que inclui elicitação, especificação e validação de requisitos.
- A elicitação de requisitos é um processo iterativo que pode ser representado como uma espiral de atividades - descoberta de requisitos, classificação e organização de requisitos, negociação de requisitos e documentação de requisitos.

Pontos chave

- Você pode usar uma variedade de técnicas para elicitación de requisitos, incluindo entrevistas e etnografia. Histórias de usuários e cenários podem ser usados para facilitar as discussões.



Pontos chave



- A especificação de requisitos é o processo de documentar formalmente os requisitos do usuário e do sistema e criar um documento de requisitos de software.
- O documento de requisitos de software é uma declaração acordada dos requisitos do sistema. Ele deve ser organizado de forma que os clientes do sistema e os desenvolvedores de software possam usá-lo.

Pontos chave



- A validação de requisitos é o processo de verificação dos requisitos de validade, consistência, integridade, realismo e verificabilidade.
- Mudanças comerciais, organizacionais e técnicas inevitavelmente levam a mudanças nos requisitos de um sistema de software. O gerenciamento de requisitos é o processo de gerenciamento e controle dessas mudanças.

Referências

- MAXIM, Bruce R.; PRESSMAN, Roger S.. Engenharia de Software: Uma abordagem profissional. 8. ed. São Paulo: Bookman, 2016. 940 p.
- SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p.

