

ENGENHARIA DE REQUISITOS



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

O que é um 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.
- Isso é inevitável, pois os requisitos podem ter uma função dupla
 - Pode ser a base para uma licitação de um contrato - portanto, deve estar aberto a interpretação;
 - Pode ser a base do próprio contrato - portanto, deve ser definido em detalhes;
 - Ambas as declarações podem ser chamadas de requisitos.

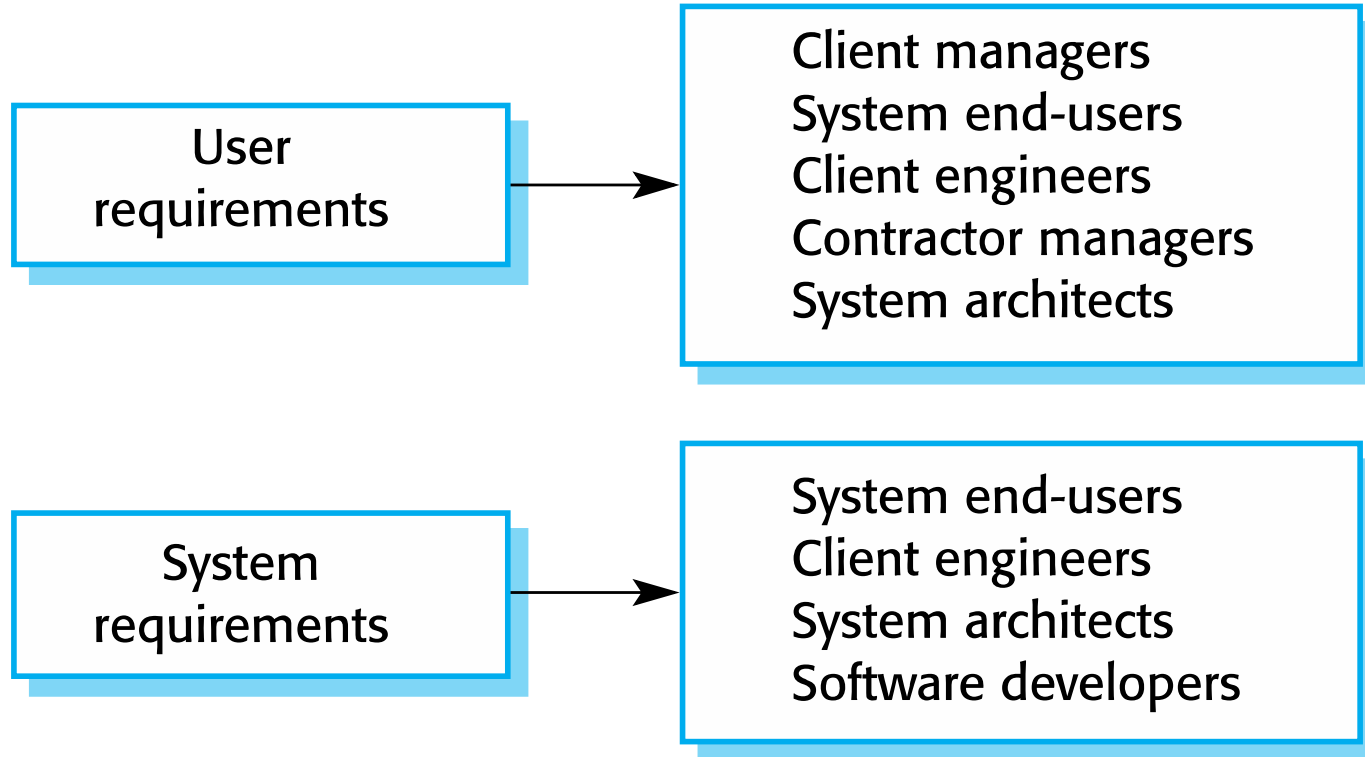
Tipos de requisitos

- Requisitos Funcionais: são declarações de serviços que o sistema deve prover, descrevendo o que o sistema deve fazer
- Requisitos não-funcionais: descrevem restrições sobre os serviços ou funções oferecidos pelo sistema

Tipos de requisitos

- Sommerville aponta dois níveis de descrição de requisitos
- 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.

Quem lê os diferentes tipos de requisitos?



Stakeholders

- 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

Tipos de requisitos

- Requisitos não funcionais têm origem
 - nas necessidades dos usuários;
 - em restrições de orçamento;
 - em políticas organizacionais;
 - em necessidades de interoperabilidade com outros sistemas de software ou hardware;
 - em fatores externos como regulamentos e legislações

Tipos de requisitos

- Sommerville classifica os requisitos não-funcionais quanto a sua origem:
 - requisitos de produto
 - requisitos organizacionais
 - requisitos externos

Tipos de requisitos

- Requisitos de produto especificam atributos de qualidade que o sistema deve apresentar:
 - Confiabilidade, usabilidade, eficiência, portabilidade, manutenibilidade e segurança

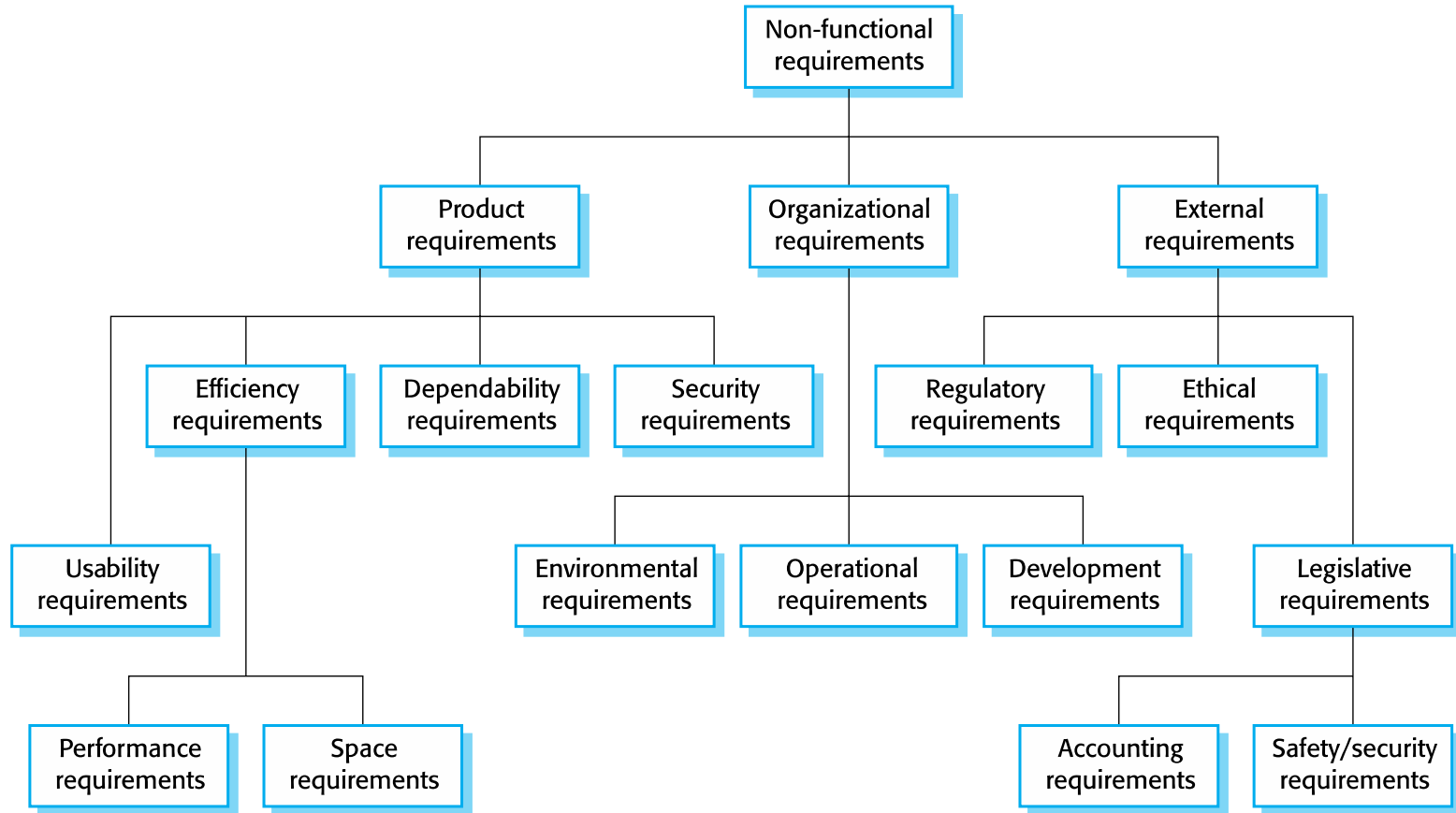
Tipos de requisitos

- Requisitos organizacionais são derivados de metas, políticas e procedimentos das organizações do cliente e do desenvolvedor. Incluem requisitos de processo, requisitos de implementação, restrições de entrega, restrições orçamentárias etc

Tipos de requisitos

- Requisitos externos referem-se a todos os requisitos derivados de fatores externos ao sistema e seu processo de desenvolvimento, tais como requisitos de interoperabilidade com sistemas de outras organizações, requisitos legais e éticos

Tipos de 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.
- 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.

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

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.
- Considere o termo "pesquisar" no requisito 1
 - Intenção do usuário - procure o nome do paciente em todas as consultas em todas as clínicas;
 - Interpretação do desenvolvedor - pesquise o nome de um paciente em uma clínica individual. O usuário escolhe a clínica e depois pesquisa.

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.

Métodos ágeis e 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

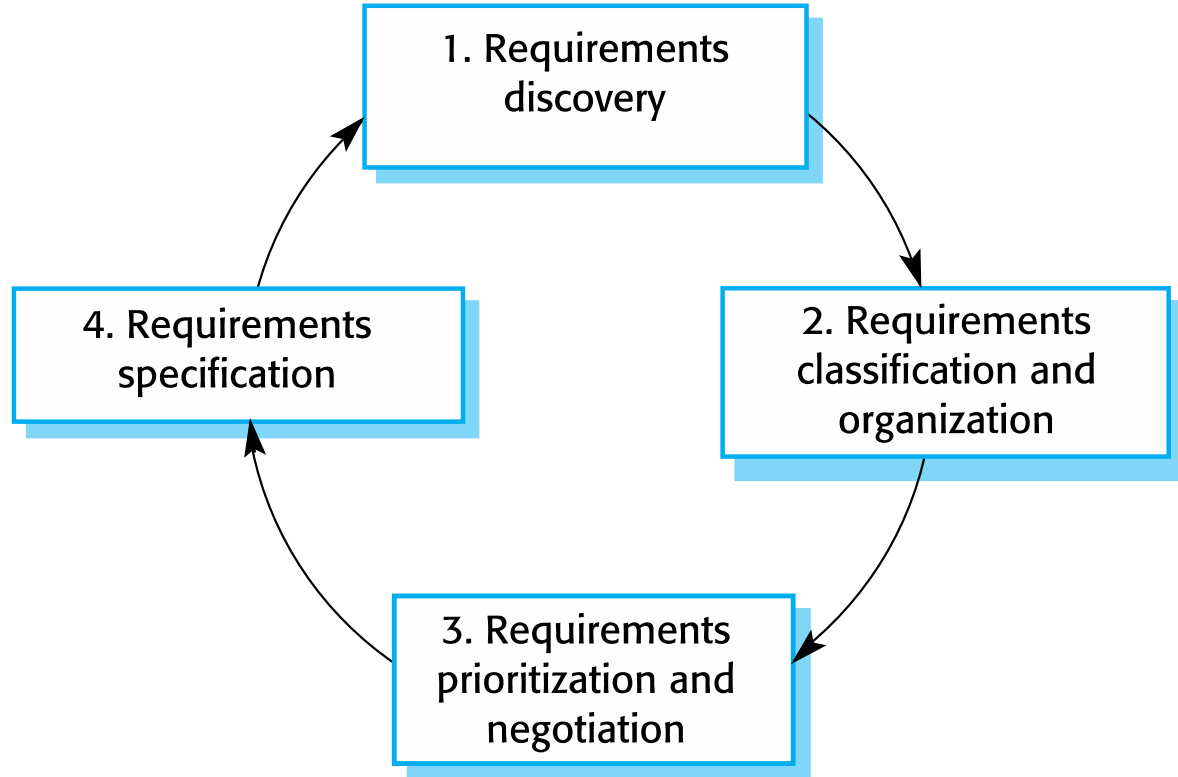
Métodos ágeis e requisitos

- Métodos ágeis geralmente usam engenharia de requisitos incrementais e podem expressar requisitos como "histórias de usuário" .

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.

Processo de Elicitação de Requisitos



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

- 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

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 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.

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.

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 r0 is replaced by r1 then r1 is replaced by r2.

Side effects None.

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

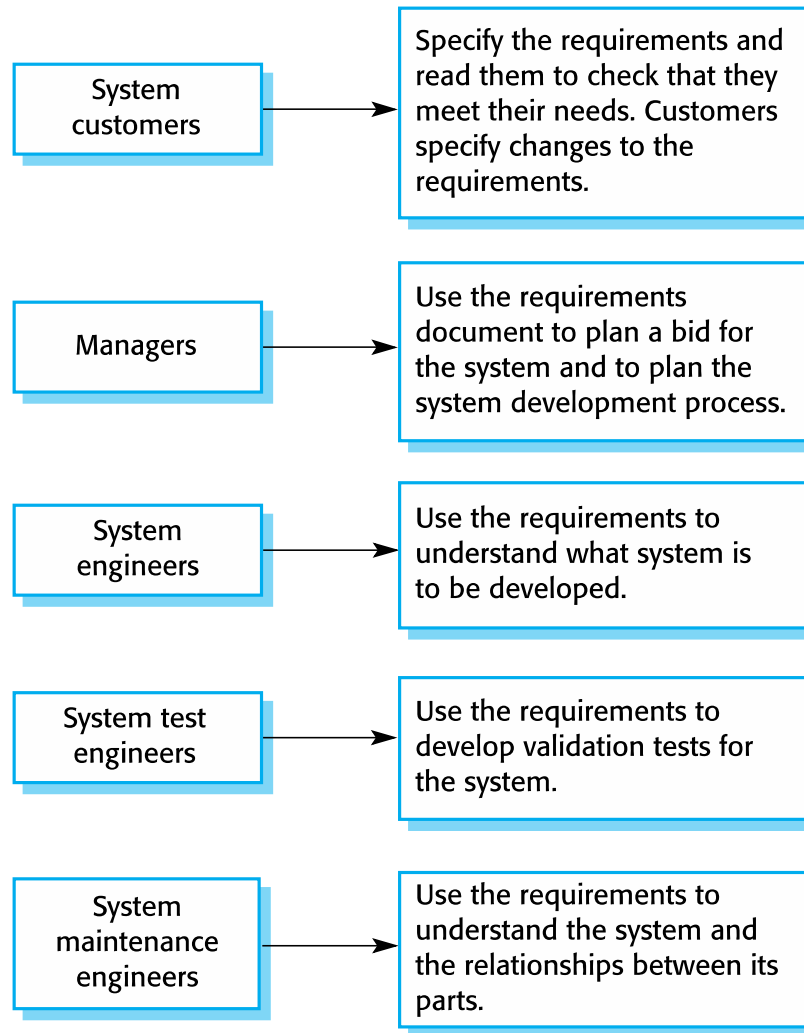
Condition	Action
Sugar level falling ($r_2 < r_1$)	CompDose = 0
Sugar level stable ($r_2 = r_1$)	CompDose = 0
Sugar level increasing and rate of increase decreasing ($(r_2 - r_1) < (r_1 - r_0)$)	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ($(r_2 - r_1) \geq (r_1 - r_0)$)	CompDose = round $((r_2 - r_1)/4)$ If rounded result = 0 then CompDose = MinimumDose

Casos de uso

- Os casos de uso são um tipo de diagrama 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.

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.



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.

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

- 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 (com as alterações consequentes no suporte do sistema necessário) e podem ser introduzidas novas legislações e regulamentos que o sistema deve necessariamente cumprir.

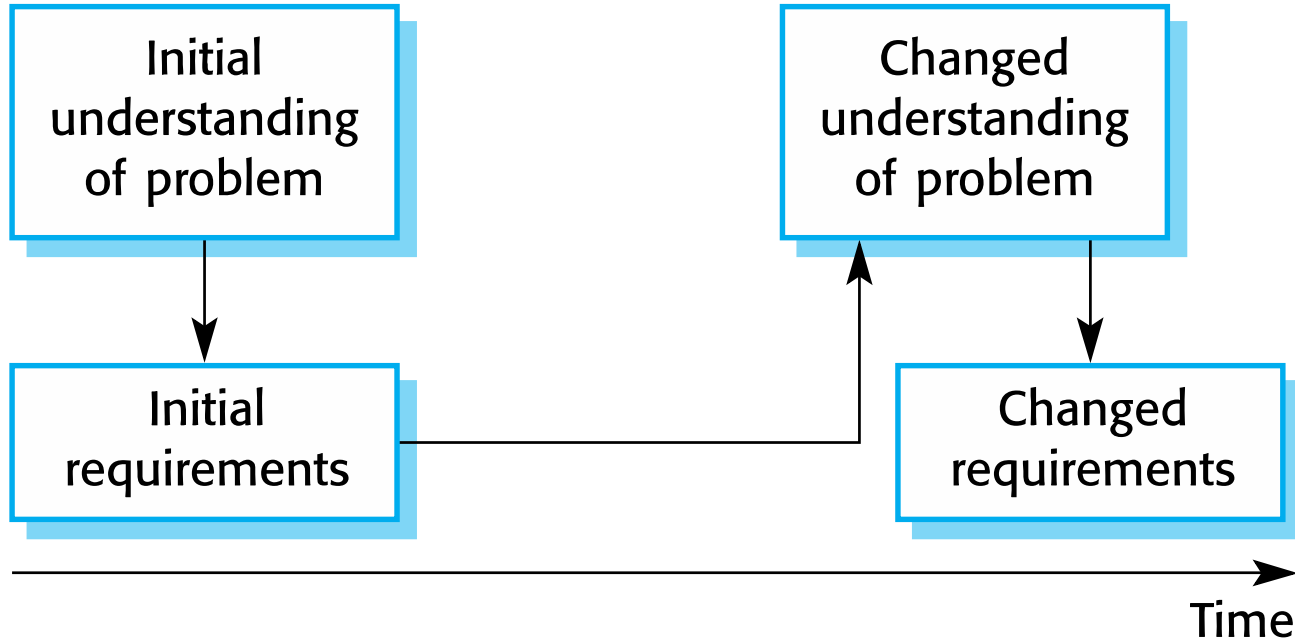
Mudança de requisitos

- 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 se o sistema deve atender aos seus objetivos.

Requisitos em mudança

- 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.

Evolução de requisitos



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.

Planejamento de mudança de requisitos

- 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.
 - 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.

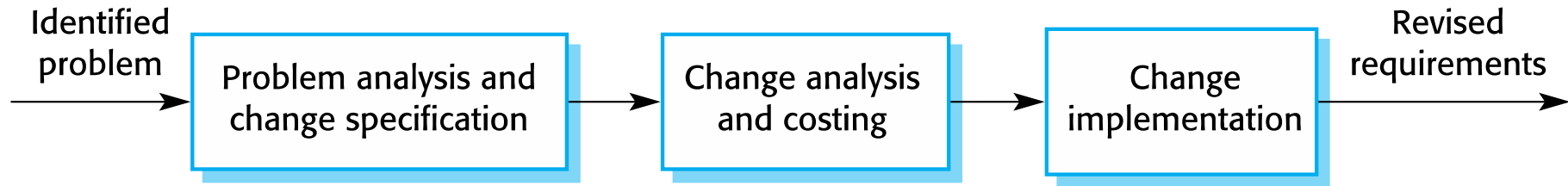
Gestão de mudança de requisitos

- 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.
- 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.

Gestão de mudança de 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

- 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.