



Engenharia de Requisitos

Metodologias e Desenvolvimento de Software

Pedro Salgueiro

pds@uevora.pt

CLV-256



- *Outline*
 - Requisitos de utilizador e de sistema;
 - Requisitos funcionais e não funcionais;
 - Documento de requisitos;
 - Especificação de requisitos;
 - Processos de engenharia de requisitos;
 - Identificação e análise de requisitos;
 - Validação de requisitos;
 - Gestão de requisitos;



- Processo
 - Enumerar
 - serviços que o cliente precisa do sistema;
 - restrições em que o sistema opera e é desenvolvido;
- Requisitos são
 - Descrições do sistema;
 - Restrições do sistema;
 - Identificados durante o processo de engenharia de requisitos;



- O que é um requisito?
 - Desde uma especificação abstrata de alto nível:
 - de um serviço;
 - de uma restrição do sistema;
 - Até à especificação de formulas matemáticas detalhadas;
 - Várias utilizações
 - Base para a negociação de um contrato de desenvolvimento;
 - Aberto a interpretações;
 - Base para o contrato de desenvolvimento;
 - Definido em detalhe;
 - Ambos são considerados requisitos;



Abstração dos requisitos (Davis)

“If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not pre-defined. The requirements must be written so that several contractors can bid for the contract, offering, perhaps, different ways of meeting the client organization’s needs. Once a contract has been awarded, the contractor must write a system definition for the client in more detail so that the client understands and can validate what the software will do. Both of these documents may be called the requirements document for the system.”



Tipos de requisitos

- Requisitos de utilizador
 - Frases escritas em língua natural;
 - Diagramas de serviços do sistema;
 - Restrições de utilização/operação;
 - Escritos para os clientes;
- Requisitos de sistema
 - Documento estruturado;
 - Descrições detalhadas das funcionalidades do sistema e dos serviços;
 - Restrições de utilização/operação;
 - Define o que deve ser implementado;
 - Pode fazer parte do contrato de desenvolvimento;



Requisitos de utilizador e de sistema

Exemplo

- **Requisitos de utilizador**

1. “O sistema deve gerar relatórios mensais com o custo dos medicamentos receitados por cada clínica, durante o mês em questão”.

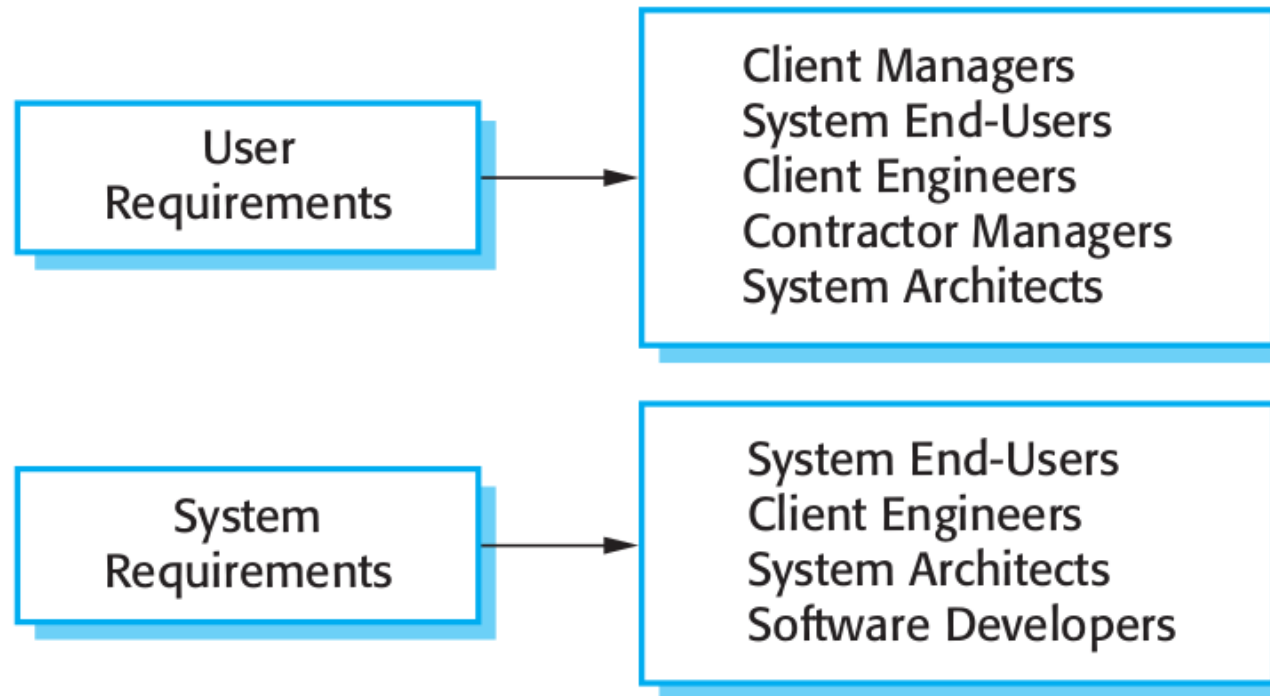
- **Requisitos do sistema**

- 1.1. “No último dia de cada mês, deve ser feito um gerado um relatório com todos os medicamentos receitados, o seu custo e a clínica que os receitou.”
- 1.2. “O sistema deve gerar automaticamente o relatório para ser impresso, depois das 17:30h, no ultimo dia do mês.”
- 1.3. “O relatório deve ser criado para cada clínica e deve listar os nomes individuais dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos receitados.”
- 1.4. “Se os medicamentos estiverem disponíveis em doses unitárias (e.g. 10mg, 20mg, etc), devem ser gerados relatórios separados para cada dose unitária.”
- 1.5. “O acesso a todos os relatórios de custos deve ser restringido apenas ao utilizadores autorizados, listados no sistema de controle de acessos.”



Requisitos de utilizador e de sistema

- Lidos por quem?
- Interessam a quem?





System *stakeholders* (interessados no sistema)

- Qualquer pessoa
 - que esteja relacionada com sistema de alguma forma;
 - que tenha algum interesse no sistema;
- Tipos de *stakeholders*
 - Utilizadores finais;
 - Gestores do sistema;
 - Donos do sistema;
 - *Stakeholders* externos;



Requisitos funcionais e não funcionais

- Requisitos funcionais
 - Descrição dos serviços que o sistema deve fornecer;
 - Descrição sobre como é que o sistema deve reagir ou comportar-se em certos inputs e condições;
 - Pode incluir o que o sistema não deve fazer;
- Requisitos não funcionais
 - Restrições sobre os serviços ou funcionalidades oferecidas pelo sistema;
 - temporais, processo de desenvolvimento, standards, etc;
 - normalmente aplicados ao sistema todo;
- Requisitos de domínio
 - Restrições ao sistema, relativamente ao domínio de aplicação do sistema;



Requisitos funcionais

- Descrevem as funcionalidades do sistema;
- Dependem de:
 - Tipo de software;
 - Utilizadores do sistema;
 - Tipo do sistema onde o software vai ser usado;
- Requisitos funcionais de utilizador;
 - Descrições de alto-nível sobre o que o sistema deve fazer;
- Requisitos funcionais do sistema;
 - Descrição detalhada sobre o que os serviços do sistema;



Requisitos funcionais - Exemplos

- “Um utilizador deve ser capaz de procurar consultas na lista de agendamentos de todas as clínicas”
- “Todos os dias, para cada clínica, o sistema deve gerar a lista de pacientes que têm consulta agendada para esse dia.”
- “Cada utilizador do sistema deve ser identificado de forma única através do seu nº de empregado composto por 8 dígitos”



Imprecisão dos requisitos

- Falta de precisão;
 - Vão surgir problemas
- Requisitos ambíguos;
 - Diferentes interpretações:
 - Por parte dos utilizadores
 - Por parte da equipa de desenvolvimento
 - Exemplo:
 - Considerando o termo o seguinte requisito: “Um utilizador deve ser capaz de procurar consultas marcadas na lista de agendamentos de todas as clínicas”
 - Utilizador: Procurar o nome de um paciente em todas as consultas de todas as clínicas
 - Equipa de desenvolvimento: Procurar o nome de um paciente numa única clínica. O utilizador escolhe a clínica onde a pesquisa deve ser feita.



Completude e consistência dos requisitos

- Devem ser consistentes e “completos”;
- Completos
 - Incluir todas as descrições;
- Consistentes
 - Não existir conflitos ou contradições entre requisitos;
- Na prática
 - Muito difícil de criar um documento de requisitos completo e consistente;



Requisitos não funcionais

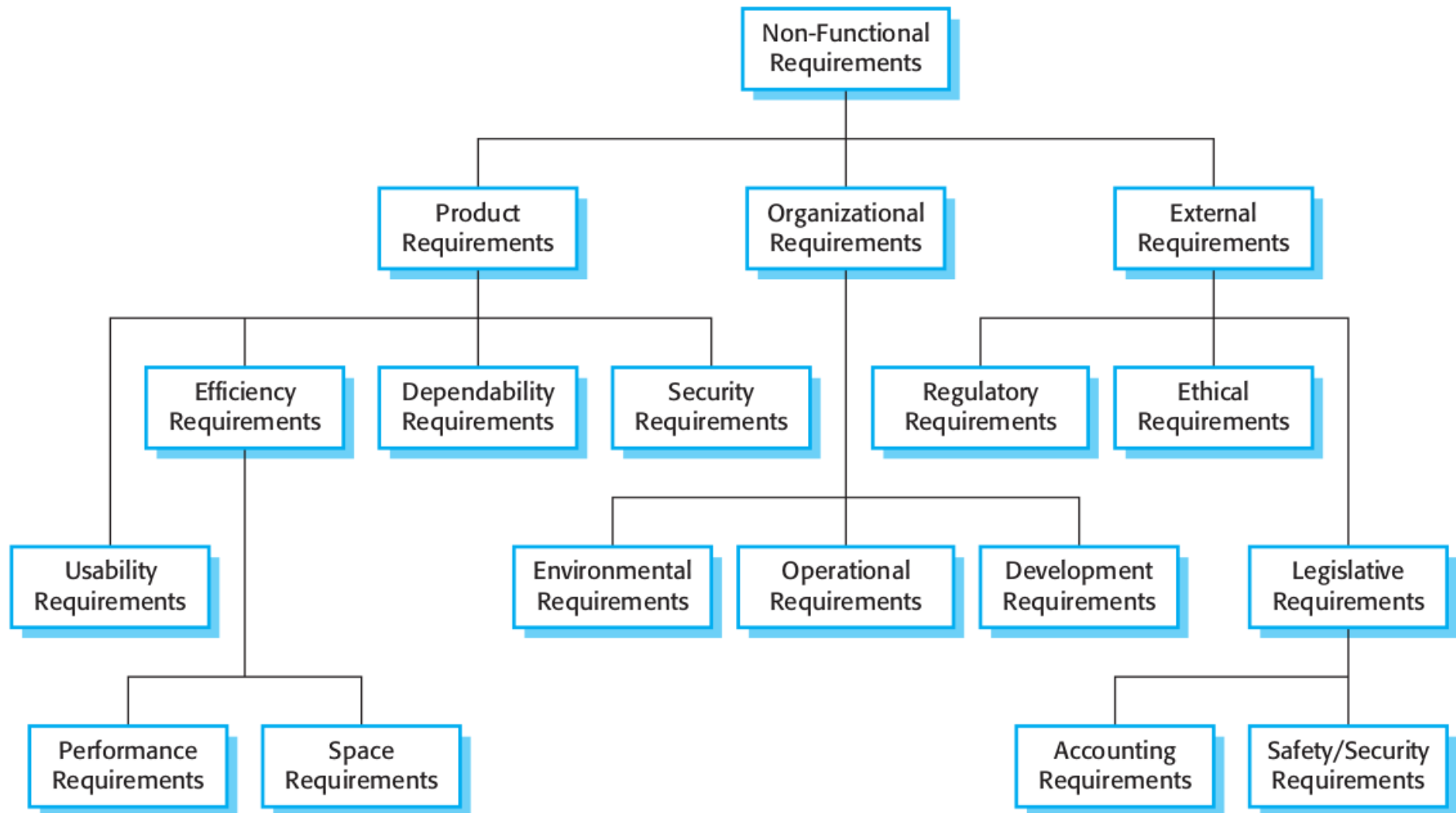
- Definem propriedades e restrições ao sistema
 - exemplos: fiabilidade, tempo de resposta requisitos de *storage*, dispositivos de I/O, etc.
- Requisitos sobre o processo de software
 - exemplos: uso de um IDE específico, de uma linguagem ou de um método de desenvolvimento
- Podem ser mais críticos que os requisitos funcionais
 - Se não forem satisfeitos, podem comprometer todo o sistema



Requisitos não funcionais

- Podem afetar o sistema de forma global
 - Exemplo
 - Requisito relacionado com o desempenho do sistema
 - Pode ser necessário organizar todo o sistema por forma a garantir a boa comunicação entre componentes
- Um requisito não funcional pode dar origem a vários requisitos funcionais
 - Exemplo
 - Um requisito não funcional relacionado com segurança, pode dar origem a diversos requisitos funcionais que garantam que o requisito não funcional é verificado.
 - Podem também introduzir restrições a requisitos já existentes

Tipos de requisitos não funcionais





Tipos de requisitos não funcionais

- Requisitos do produto
 - Requisitos que especificam qual o comportamento do sistema
 - tempo de execução, fiabilidade, etc...
- Requisitos organizacionais
 - Requisitos que resultam das *políticas* e procedimentos da organização/empresa:
 - processos usados, requisitos de implementação, etc...
- Requisitos externos
 - Requisitos relacionados com factores externos ao sistema
 - requisitos de interoperabilidade, requisitos legais, etc...



Requisitos não funcionais - Exemplos

- Requisito de produto:
 - O sistema deve estar disponível para todas as clínicas, durante as horas normais de trabalho (Seg-Sex, 08:30h – 17:30h). O sistema não pode estar em baixo mais do que 5 segundos por dia, em qualquer dia.
- Requisito organizacional:
 - Os utilizadores do sistema devem ser capazes de se autenticar usando o seu cartão de identificação.
- Requisito externo:
 - O sistema deve implementar as regras de privacidade especificadas no documento HStan-03-2006-priv.



Objetivos (*goals*) e requisitos

- Problema
 - Requisitos não funcionais são muito abstratos e imprecisos (normalmente);
 - Difíceis de verificar se estão satisfeitos;
- Solução
 - Criar *goals* (objetivos)
 - Uma descrição genérica sobre a intenção do utilizador
 - Exemplo: “Facilidade de utilização”
 - **E** associar aos *goals* requisitos não funcionais verificáveis
 - Uma descrição que use algo que possa ser medido e testado de forma objetiva
- *Goals*
 - Úteis para a equipa de desenvolvimento. Definem as intenções de uso dos utilizadores.



Exemplo - Requisito de usabilidade

- *Goal*
 - “O sistema deve ser fácil de usar por parte da equipa médica e deve estar organizado de forma a que os erros feitos pelos utilizadores sejam minimizados.”
- Requisito não funcional verificável
 - “A equipa médica deve conseguir usar todas as funcionalidades do sistema após 4 horas de formação. Após esta formação, o número médio de erros cometidos pelos utilizadores não deve ser superior a 2 por cada hora de utilização 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



Processos de Engenharia de Requisitos

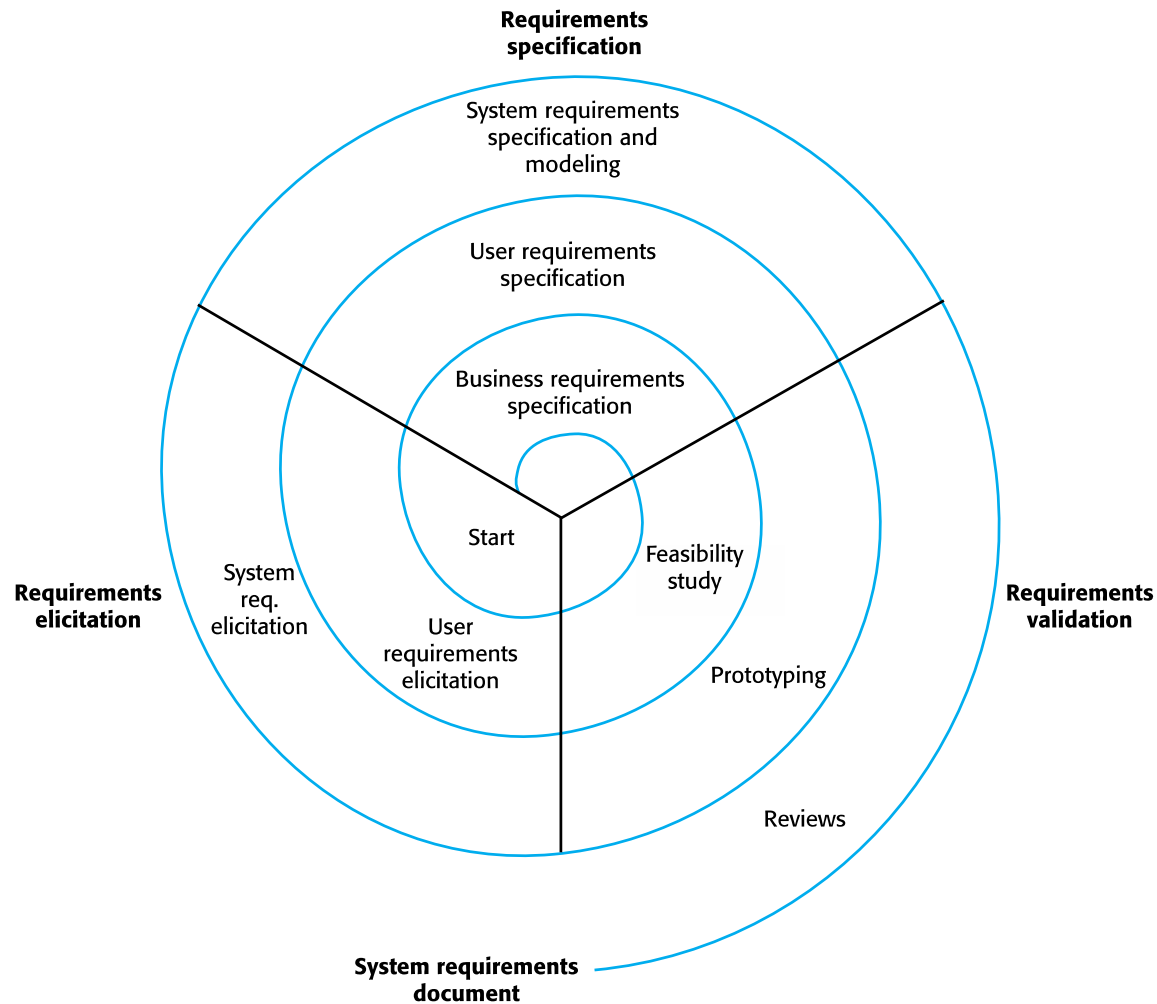


Processos de engenharia de requisitos

- Variam dependendo de:
 - Domínio de aplicação;
 - Pessoas envolvidas;
 - Entidade que desenvolve os requisitos;
- No entanto, existem atividades comuns a todos os processos:
 - Identificação dos requisitos;
 - Análise dos requisitos;
 - Validação dos requisitos;
 - Gestão dos requisitos;
- Na prática
 - Atividades feitas de forma iterativa e intercalada;



Processos de engenharia de requisitos





Identificação dos requisitos



Identificação e análise de requisitos

- Também conhecido por descoberta de requisitos;
- Trabalho conjunto da equipa técnica com os clientes por forma a descobrir:
 - o domínio da aplicação;
 - os serviços que devem ser fornecidos;
 - restrições de operação/utilização;
- Pode envolver todos os *stakeholders*:
 - Utilizadores finais;
 - Gestores;
 - Engenheiros envolvidos na manutenção;
 - Especialistas de domínio;
 - Etc...



Identificação dos requisitos

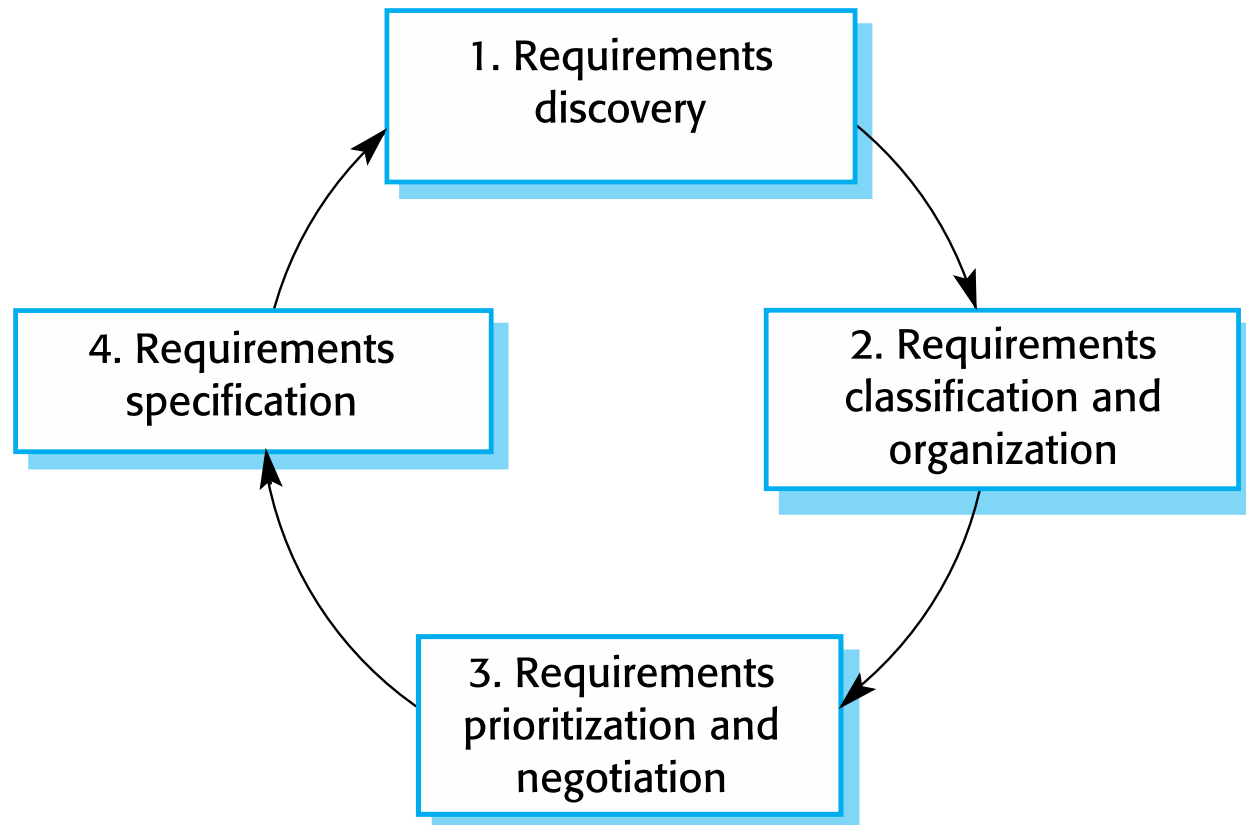
- Engenheiros trabalham com os *stakeholders* por forma descobrir:
 - Domínios de aplicação;
 - Que serviços devem ser fornecidos;
 - Desempenho do sistema;
 - Restrições de hardware;
 - Etc...;
- Deve incluir
 - Descoberta dos requisitos;
 - Classificação e organização dos requisitos;
 - Prioritização e negociação dos requisitos;
 - Especificação dos requisitos;



Problemas na identificação dos requisitos

- *Stakeholders* não sabem o que realmente querem;
- *Stakeholders* expressam os requisitos nos seus termos próprios;
- Diferentes *stakeholders* podem ter requisitos com conflitos;
- Fatores organizacionais e políticos podem influenciar os requisitos de sistema;
- Os requisitos mudam durante o processo de análise:
 - Podem surgir novos *stakeholders*;
 - As regras de negócio podem alterar-se;

Processo de identificação e análise dos requisitos





Identificação dos requisitos – Atividades

- Descoberta de requisitos
 - Interagir com todos os *stakeholders* por forma a descobrir os requisitos
 - Requisitos de domínio são também identificados
 - Fazer a distinção entre requisitos de utilizador e de sistema
- Classificação e organização dos requisitos
 - Agrupar requisitos relacionados
 - Organizá-los em conjuntos coerentes
- Prioritização e negociação
 - Atribuir prioridades
 - Resolver conflitos entre requisitos
- Especificação dos requisitos
 - Escrita dos requisitos



Histórias e cenários

- Exemplos reais sobre como o sistema pode ser usado
- Descrição sobre como o sistema pode ser usado para realizar uma tarefa específica
- Importantes para os *stakeholders*
 - Identificam-se facilmente com o cenário
 - Apresentam um caso prático
 - Conseguem facilmente comentar o cenário



Histórias e cenários - Exemplo

Jack is a primary school teacher in Ullapool (a village in northern Scotland). He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing. As part of this, pupils are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history resources site) to access newspaper archives and photographs. However, Jack also needs a photo sharing site as he wants pupils to take and comment on each others' photos and to upload scans of old photographs that they may have in their families.

Jack sends an email to a primary school teachers group, which he is a member of to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account. He uses the iLearn setup service to add KidsTakePics to the services seen by the pupils in his class so that when they log in, they can immediately use the system to upload photos from their mobile devices and class computers.



Cenários

- *User story* estruturada
- Devem incluir:
 - Descrição da situação inicial (ou ponto de partida);
 - Descrição do fluxo normal de eventos;
 - Descrição do que pode correr mal;
 - Informação sobre outras atividades concorrentes;
 - Descrição do estado do sistema quando o cenário termina;



Cenários

- Exemplo: Fazer upload de fotografias/imagens
 - **Estado inicial:** “A user or a group of users have one or more digital photographs to be uploaded to the picture sharing site. These are saved on either a tablet or laptop computer. They have successfully logged on to KidsTakePics.”
 - **Comportamento normal:**
 - “The user chooses upload photos and they are prompted to select the photos to be uploaded on their computer and to select the project name under which the photos will be stored. They should also be given the option of inputting keywords that should be associated with each uploaded photo. Uploaded photos are named by creating a conjunction of the user name with the filename of the photo on the local computer.”
 - “On completion of the upload, the system automatically sends an email to the project moderator asking them to check new content and generates an on-screen message to the user that this has been done.”



Cenários

- Exemplo: Fazer upload *de fotografias*
 - **O que pode correr mal:**
 - “No moderator is associated with the selected project. An email is automatically generated to the school administrator asking them to nominate a project moderator. Users should be informed that there could be a delay in making their photos visible.”
 - “Photos with the same name have already been uploaded by the same user. The user should be asked if they wish to re-upload the photos with the same name, rename the photos or cancel the upload. If they chose to re-upload the photos, the originals are overwritten. If they chose to rename the photos, a new name is automatically generated by adding a number to the existing file name.”
 - **Outras atividades:** “The moderator may be logged on to the system and may approve photos as they are uploaded.”
 - **Estado final depois de terminar:** “User is logged on. The selected photos have been uploaded and assigned a status ‘awaiting moderation’. Photos are visible to the moderator and to the user who uploaded them.”



Especificação de requisitos



Especificação de requisitos

- Processo de escrever os requisitos
 - requisitos de sistema e de utilizador
 - documento de requisitos
- Requisitos devem ser de fácil compreensão
 - pelos utilizadores finais
 - pelos clientes que não têm um *background* técnico
- Requisitos de sistema
 - mais detalhados
 - podem incluir alguns aspetos técnicos
- Os requisitos podem fazer parte de um contrato para desenvolvimento do sistema
 - Devem ser o mais completos possível



Formas de escrever os 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 desenho do sistema

- Preferencialmente
 - Requisitos especificam o que o sistema deve fazer;
 - Desenho descreve como é feito;
- Na prática, requisitos e desenho são inseparáveis
 - A arquitetura do sistema pode ser desenhada para estruturar os requisitos;
 - O sistema pode interagir com outros sistemas que geram requisitos de desenho;
 - O uso de arquiteturas específicas pode satisfazer requisitos não-funcionais, associados a requisitos de domínio;



Especificação em língua natural

- Requisitos são escritos em língua natural;
 - Complementados com diagramas e tabelas;
- Porquê o uso de língua natural:
 - Expressiva
 - Intuitiva
 - Universal
 - Requisitos podem ser lidos/percebidos por clientes e utilizadores



Escrita dos requisitos – Guião

- “Criar” um formato *standard* para todos os requisitos;
- Usar linguagem de forma consistente:
 - Usar “deve” para requisitos obrigatórios
 - Usar “pode” para requisitos desejáveis
- Usar *text highlight* para identificar aspetos importantes do requisito;
- Evitar termos informáticos/técnicos;
- Incluir uma explicação sobre o razão da necessidade do requisito ser necessário;



Língua Natural - Problemas

- Falta de clareza:
 - Não é fácil ser preciso sem tornar o documento difícil de ler;
- Confusão de requisitos:
 - Os requisitos funcionais e não funcionais tendem a ficar misturados;
- Junção de requisitos:
 - Vários requisitos (diferentes) podem ser descritos no mesmo requisito;



Especificações estruturadas

- Abordagem para escrever requisitos:
 - Liberdade limitada para descrever os requisitos;
 - Escritos seguindo um padrão específico;
- Funciona bem para alguns tipos de sistema:
 - e.g.: sistemas de controle *embedded*/embarcados;
 - muito rígidos para grande parte dos sistemas;



Especificações baseadas em formulários

- Definição da funcionalidade ou entidade;
- Descrição dos inputs e qual a sua origem;
- Descrição dos outputs e qual o seu destino;
- Informação sobre a dados/informação e outras entidades necessárias;
- Descrição das ações a serem tomadas;
- Pré e pós condições (se existirem);
- Efeitos secundários (se existirem);



Especificações baseadas em formulários - Exemplo

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.



Especificações baseadas em formulários - Exemplo

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.



Especificação tabular

- Usada para complementar a língua natural;
- Particularmente útil;
 - Necessário especificar várias alternativas possíveis

Especificações tabular - Exemplo

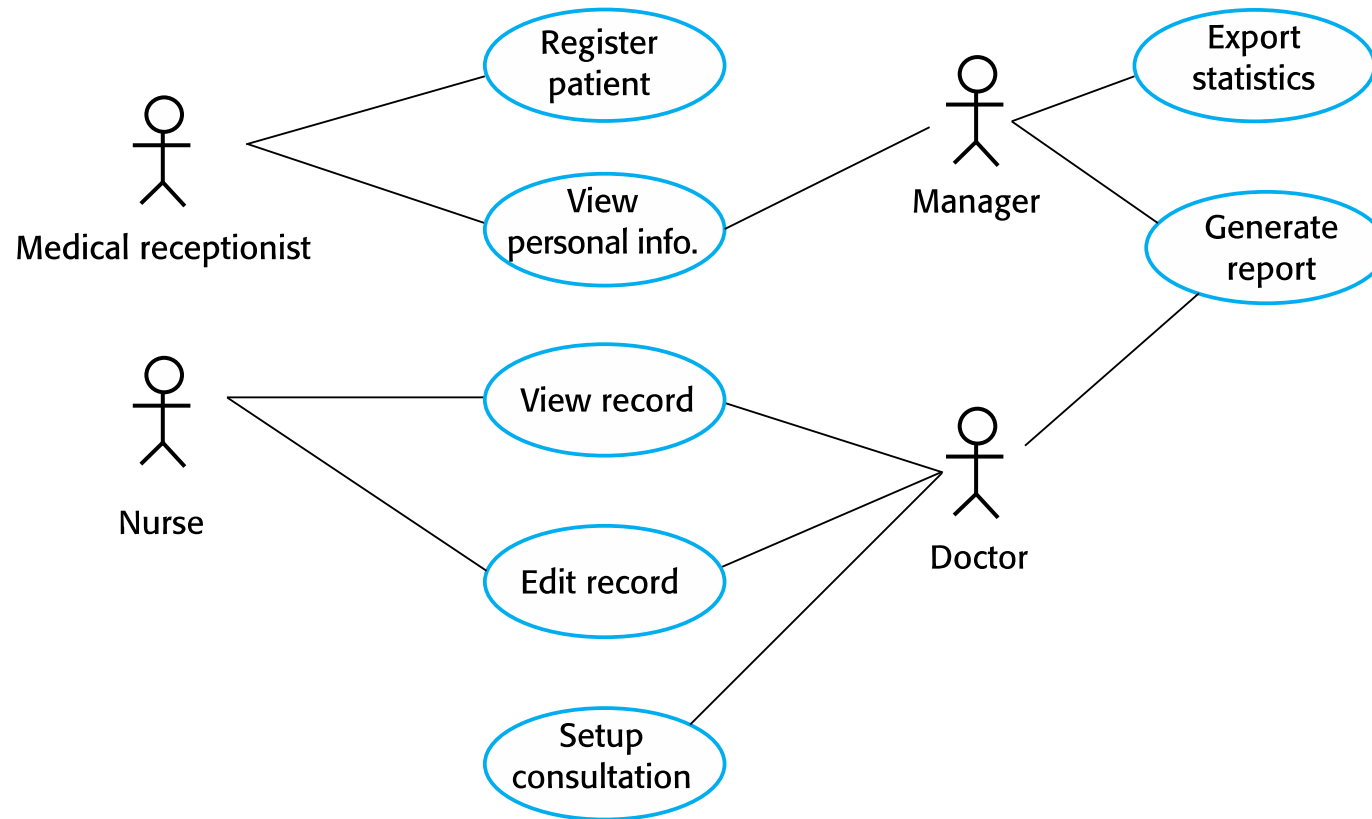
Condition	Action
Sugar level falling ($r2 < r1$)	CompDose = 0
Sugar level stable ($r2 = r1$)	CompDose = 0
Sugar level increasing and rate of increase decreasing ($(r2 - r1) < (r1 - r0)$)	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ($(r2 - r1) \geq (r1 - r0)$)	CompDose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



Use Cases

- Tipo de cenários;
 - Fazem parte de UML;
- Identificam:
 - Atores;
 - Numa interação com o sistema;
- Modelo gráfico de alto nível;
 - Complementado por uma descrição tabular;
- Diagramas de sequência (UML):
 - Podem ser usados para detalhar os use cases;
 - Mostram a sequência de eventos num sistema;

Use Cases - Exemplo



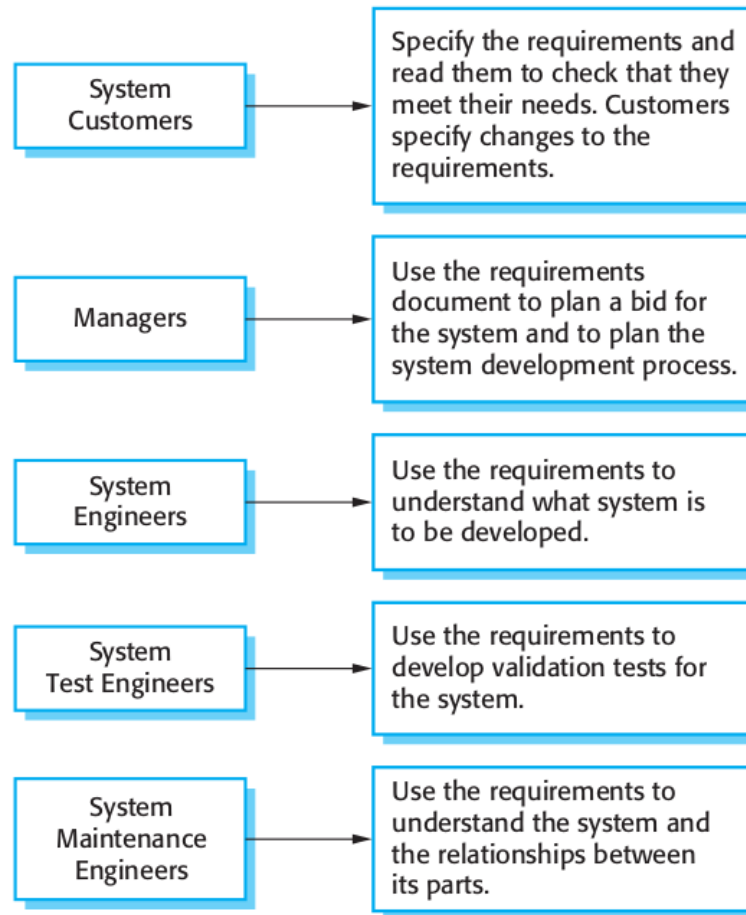


Documento de requisitos

- Documento oficial que indicia à equipa de desenvolvimento o que é esperado do sistema;
- Deve incluir a definição dos requisitos de utilizador bem como os requisitos do sistema;
- **Não é um documento de desenho**
 - Deve dizer o que o sistema deve fazer
 - **Nunca dizer como deve ser feito**



Utilizações do documento de requisitos





Documento de requisitos - Formato

- Não existe um formato único
- Informação no documento de requisitos depende do tipo de sistema a ser desenvolvido e da abordagem de desenvolvimento
- Sistemas desenvolvidos de forma incremental, tipicamente, têm um documento de requisitos menos detalhado
- Existem alguns standards para o documento de requisitos
 - IEEE
 - Usados apenas a projetos grandes

Documento de requisitos – formato IEEE

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.

Documento de requisitos – formato IEEE

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 dos requisitos

- Demonstrar que os requisitos definem o sistema que se pretende implementar;
 - Custos de corrigir requisitos depois do sistema estar terminado podem ser até 100 vezes superiores ao custo de corrigir um problema de implementação;
- Como validar
 - Validade dos requisitos;
 - Consistência;
 - Completude;
 - Realismo;
 - Verificabilidade;



Gestão dos requisitos

- Processo de gerir as alterações dos requisitos durante todo o processo
 - Aparecimento de novos requisitos;
 - Gerir dependências entre requisitos;
 - Saber quais os impactos de alterar um requisito
- Estabelecer um processo formal
 - efetuar alterações nos requisitos
 - estabelecer ligações entre requisitos



- *Software Engineering*. Ian Sommerville. 10th Edition. Addison-Wesley. 2016. Capítulo 4.