



## Engenharia de Requisitos

Metodologias e Desenvolvimento de Software

**Pedro Salgueiro**

pds@di.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



## Tipos de requisitos

- Requisitos de utilizador
  - Funcionalidades que o sistema fornece
  - Restrições de operação/funcionamento do sistema
  - Escritos em língua natural
    - Para o cliente
- Requisitos de sistema
  - Documento estruturado
  - Descrições detalhadas
    - funcionalidades do sistema
    - restrições operacionais
  - Definem o que deve ser implementado
    - Pode fazer parte de um “contrato” de implementação



## Requisitos de utilizador e de sistema

### Exemplo

- **Requisitos de utilizador**

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

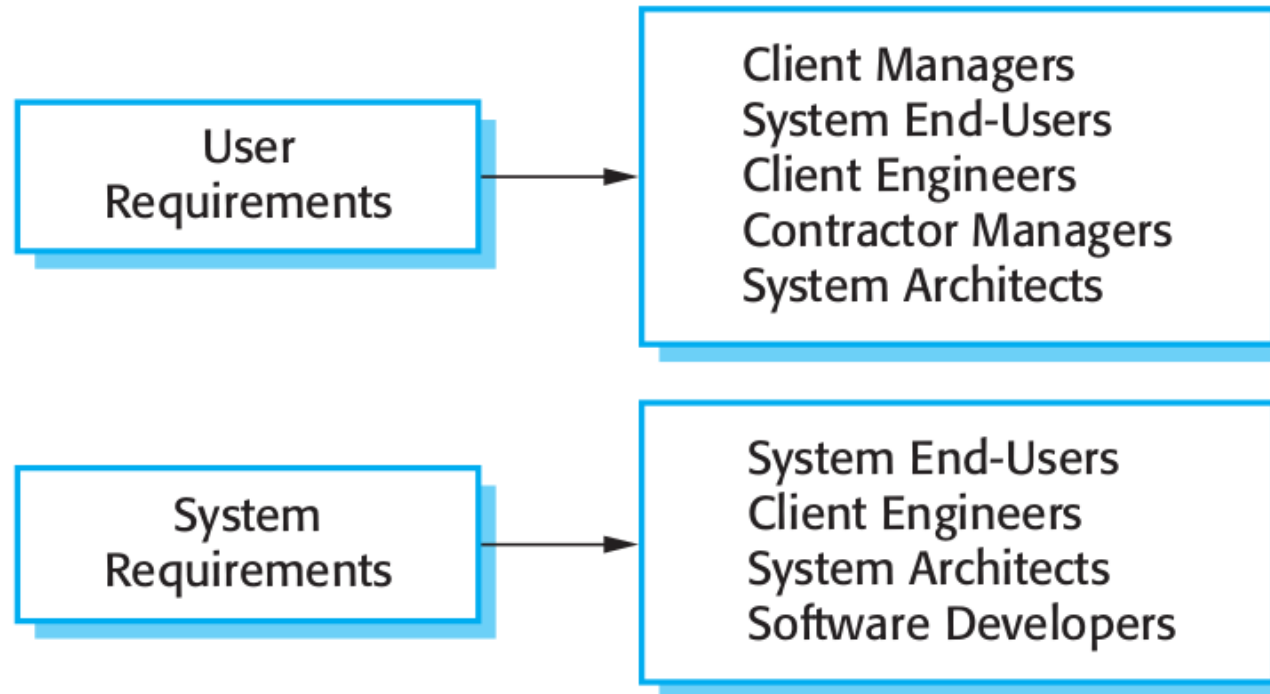
- **Requisitos do sistema**

- “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.”
- “O sistema deve gerar automaticamente o relatório para ser impresso, depois das 17:30h, no ultimo dia do mês.”
- “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.”
- “Se os medicamentos estiverem disponíveis em doses unitárias (e.g. 10mg, 20mg, etc), devem ser gerados relatórios separados para cada dose unitária.”
- “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?





## Requisitos funcionais e não funcionais

- Requisitos funcionais
  - Descrição dos serviços que o sistema deve fornecer
  - Como é que o sistema deve reagir ou comportar-se a 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 a lista de agendamentos para todas as clínicas”
- “Todos os dias, para cada clínica, o sistema deve gerar a lista de pacientes que têm consulta agendada para o 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 “pesquisar” no exemplo anterior
    - 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.



## Compleitude 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

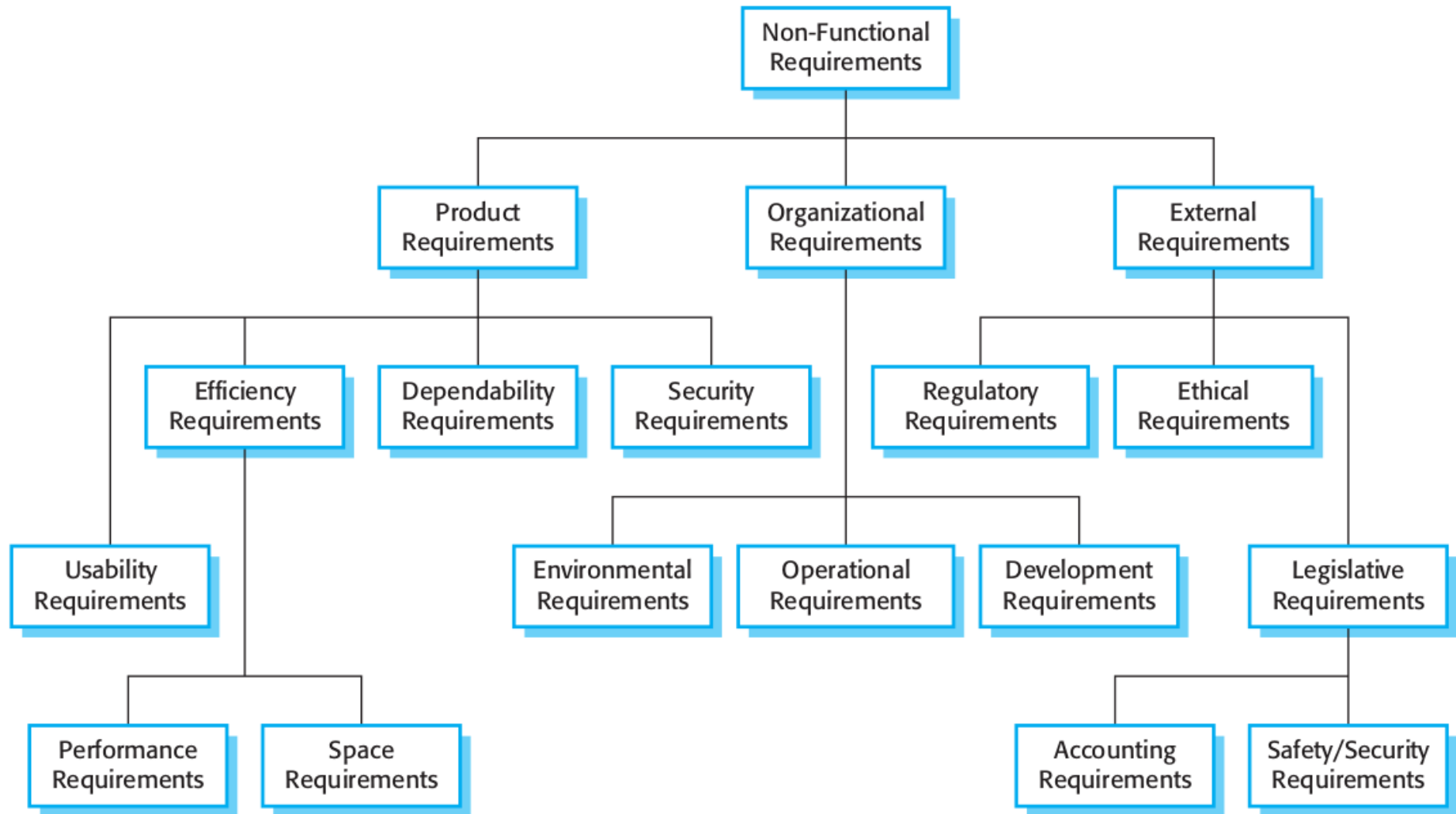
- Definem propriedades e restrições ao sistema
  - exemplos: fiabilidade, tempo de resposta requisitos de *storage*, dispositivos de I/O, etc.
- Requisitos sobre o processo de software
  - exemplos: uso de um IDE específico, de uma linguagem ou de um método de desenvolvimento
- Podem ser mais críticos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



## Requisitos não funcionais

- Afectam 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





## Requisitos não funcionais

- **Classes**
  - 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.





## Objectivos e requisitos

- Problema
  - Requisitos não funcionais são muito abstractos e imprecisos (normalmente)
  - Difíceis de verificar se estão satisfeitos
- Solução
  - Criar *goals* (objectivos)
    - Uma descrição genérica sobre a intenção do utilizador
    - Exemplo: “Facilidade de utilização”
  - e associar requisitos não funcionais verificáveis aos *goals*
    - Uma descrição que use algo que possa ser medido e testado de forma objectiva
- *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.”



## Requisitos de domínio

- Requisitos impostos pelo domínio de utilização/operação do sistema
  - Exemplo: Um sistema de controlo de um comboio tem de ter em considerações as características de travagem em diferentes condições climáticas
- Podem dar origem a
  - Novos requisitos funcionais
  - Restrições sobre requisitos já existentes
  - Definir condições especiais
- Importância
  - Se não forem satisfeitos, o sistema pode tornar-se num sistema não usável



## Requisitos de domínio

---

- Problemas
  - Compreensão
    - Requisitos são expressos usando a linguagem do domínio de aplicação;
    - São difíceis de perceber pelos engenheiros de software que estão a desenvolver o sistema;
  - “É implícito”
    - Especialistas do domínio de aplicação percebem o domínio de tal forma, que escrevem os requisitos de domínio de forma pouco explícita;



## 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**

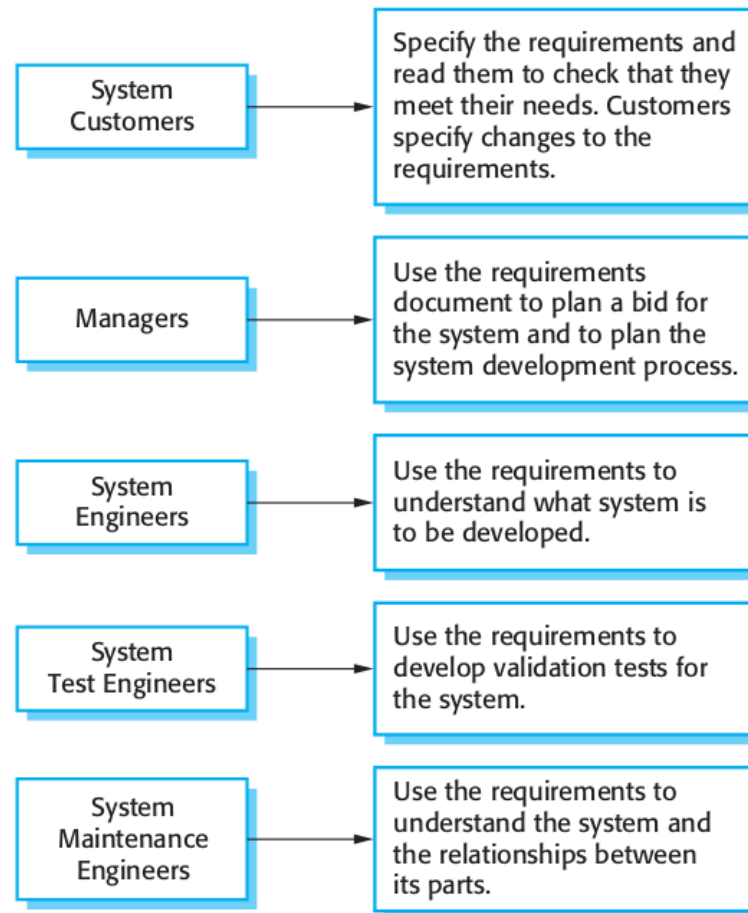


## Requisitos e metodologias ágeis

- Muitas metodologias defendem que escrever documentos de requisitos é uma perda de tempo, pois os requisitos mudam muito rapidamente
  - O documento está sempre desatualizado
- Métodos como XP (eXtreme programming) usa requisitos incrementais que são expressidos através de *user stories*
- Pouco prático para sistemas que requerem muita análise, como sistemas críticos, ou para sistemas desenvolvidos por várias equipas



## 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 projectos 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.



## Especificação de requisitos

- Processo de escrever os requisitos de utilizador e os requisitos de sistema num documento de requisitos
- Os requisitos devem ser de fácil compreensão pelos utilizadores finais e pelos clientes que não têm um *background* técnico
- Os requisitos de sistema são mais detalhados e devem poder incluir alguns aspectos técnicos
- Os requisitos podem parte de um contrato
  - Devem ser o mais completos possível



## Requisitos e desenho de software

- Actividades separadas
  - Requisitos devem descrever o que deve ser feito
  - Desenho deve descrever como se faz
  
- Na prática requisitos e desenho de software são inseparáveis
  - A arquitectura do sistema pode ser desenhada para estruturar os requisitos
  - O sistema pode interagir com outros sistemas que geram parte do desenho
  - O uso de uma arquitectura específica por forma a satisfazer um requisito não funcional ou um requisito de domínio



## Escrita dos requisitos

- Requisitos são escritos usando frases em língua natural, podendo ser complementados com diagramas e tabelas
- Forma expressiva, intuitiva e universal de escrever requisitos.
  - Podem ser lidos e percebidos pelos clientes e pelo utilizadores
- Problemas
  - Falta de clareza
    - Difícil de criar documentos precisos sem torna-lo difícil de ler
  - Confusão de requisitos
    - Requisitos funcionais e não funcionais tendem a misturar-se
  - Junção de requisitos
    - Vários requisitos escritos como apenas um requisito



## Escrita dos requisitos

- Boas práticas
  - Criar um formato standard e usa-lo em todos os requisitos
  - Usar a linguagem de forma consistente:
    - Usar a palavra “deve” para requisitos obrigatórios
    - Usar a palavra “pode” para requisitos desejáveis ou opcionais
  - Usar *highlights* para identificar partes importantes do requisito
  - Evitar linguagem técnica
  - Incluir uma explicação sobre a razão do requisito ser necessário



## Escrita dos requisitos - exemplo

3.2 - O sistema deve medir a quantidade de açúcar no sangue e administrar a insulina, se necessário, a cada 10 minutos. (Alterações do nível de açúcar no sangue são lentas, não sendo necessário medições mais frequentes; medições menos frequentes podem levar a altos níveis de açúcar no sangue.)

3.6 - O sistema deve fazer testes a cada minuto de acordo com as condições a serem testadas e com as ações definidas na Tabela 1. (Um teste de rotina pode encontrar problemas de hardware e software e alertar os utilizadores que a utilização normal do sistema pode ser impossível.)

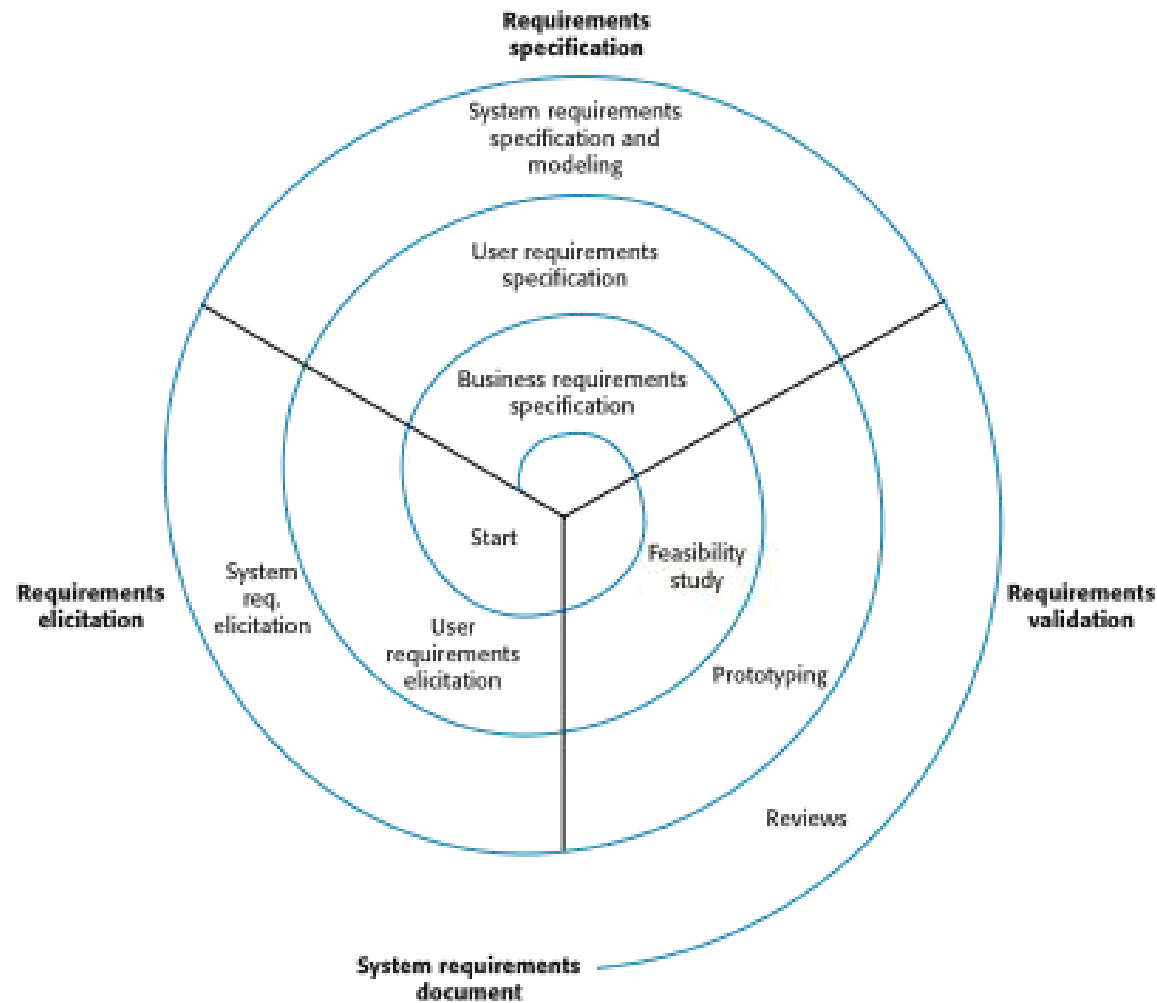


## Processo de engenharia de requisitos

- Depende de
  - Domínio da aplicação;
  - Pessoas envolvidas;
  - Empresa/organização;
- Actividades genéricas comuns
  - Identificação de requisitos;
  - Análise dos requisitos;
  - Validação dos requisitos;
  - Gestão dos requisitos;



## Espiral do processo de engenharia de requisitos





## Análise de requisitos

---

### – Problemas

- Interessados não sabem o que realmente querem;
- Interessados descrevem os requisitos usando os seus termos;
- Diferentes interessados podem ter requisitos incompatíveis entre si;
- Alteração dos requisitos durante o processo de análise;
- Factores “políticos” e organizacionais podem influenciar os requisitos;



## Identificação e análise dos requisitos

- Trabalho em conjunto
  - Engenheiros de software;
  - Interessados no sistema;
- Objectivo de encontrar
  - Domínio de aplicação do sistema;
  - Serviços que devem ser disponibilizados pelo sistema;
  - Desempenho do sistema;
  - Restrições de hardware;



## Identificação e análise dos requisitos

---

- Etapas
  - Descoberta dos requisitos;
  - Classificação e organização;
  - Prioritização e negociação dos requisitos;
  - Especificação dos requisitos;



## 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 o todo o processo
  - Aparecimento de novos requisitos;
  - Gerir dependências entre requisitos;
    - Saber quais os impactos de alterar um requisito
- Estabelecer processo formal
  - efectuar alterações nos requisitos
  - estabelecer ligações entre requisitos



- **Software Engineering, Ian Sommerville, 9th Edition, Addison-Wesley, 2010. Capítulo 4.**