TP 02
Determinação de requisitos (parte 1)

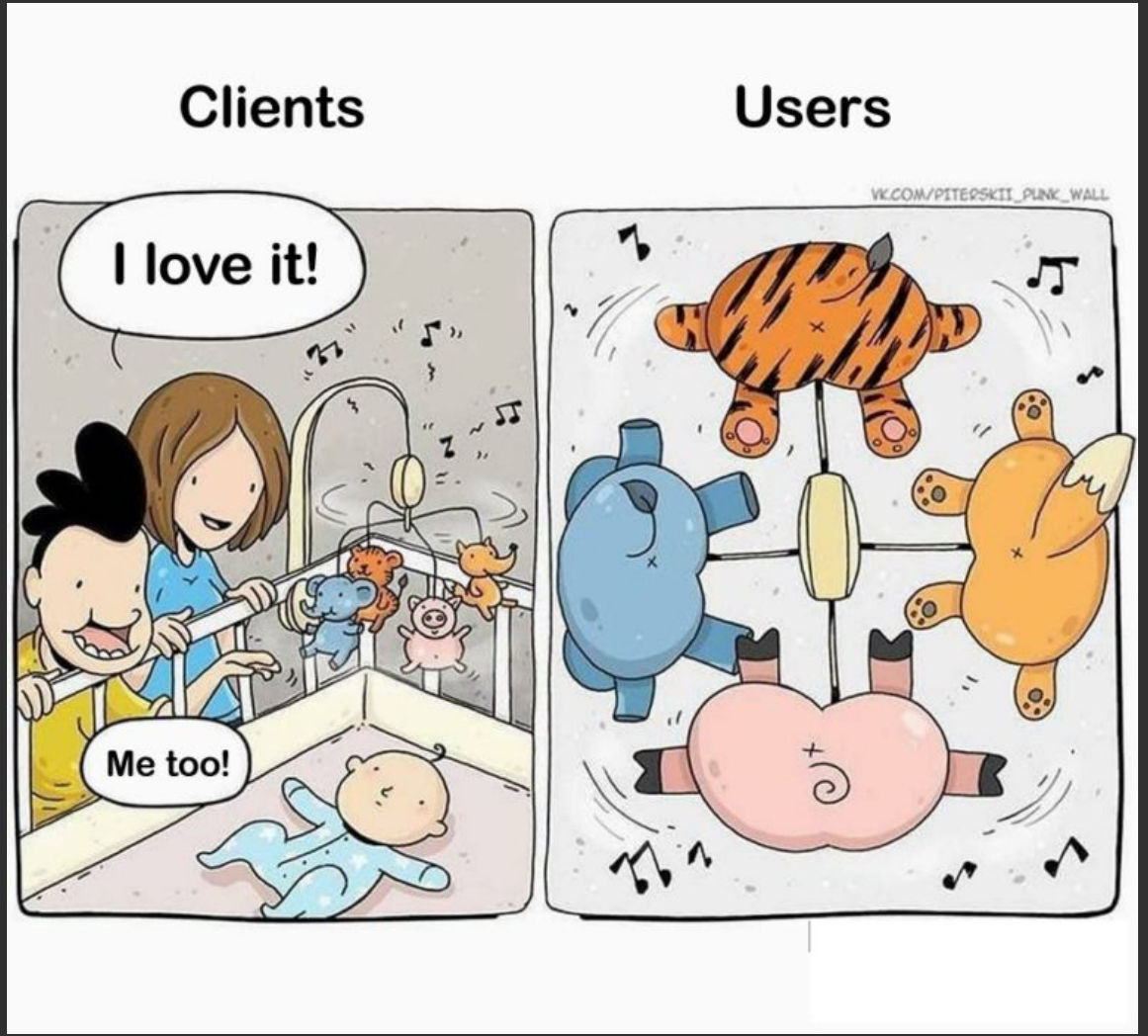
Requisitos: levantamento e análise com casos de utilização

v2025/02/18, ico@ua.pt

Objetivos de aprendizagem

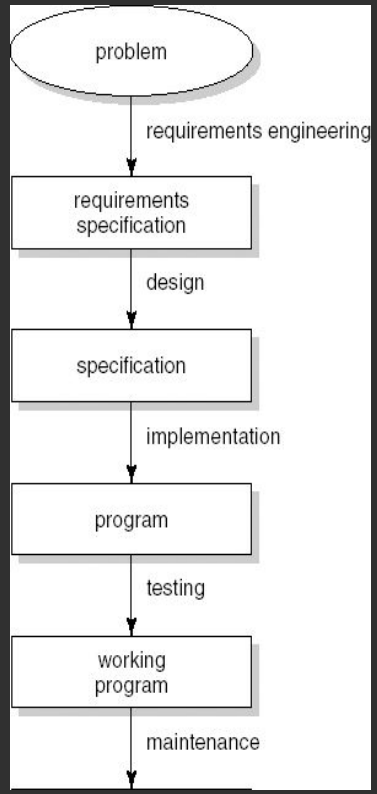
- Distinguir requisitos funcionais e não funcionais
- Enumerar técnicas de recolha de requisitos e argumentar quando usar cada uma
- Descrever técnicas de documentação de requisitos
- Explicar a relação entre requisitos e casos de utilização
- Identificar as atividades e disciplinas relacionadas com os requisitos no OpenUP

Stakeholder conflicts...



https://www.designershumor.com/2019/05/21/ux-is-important/

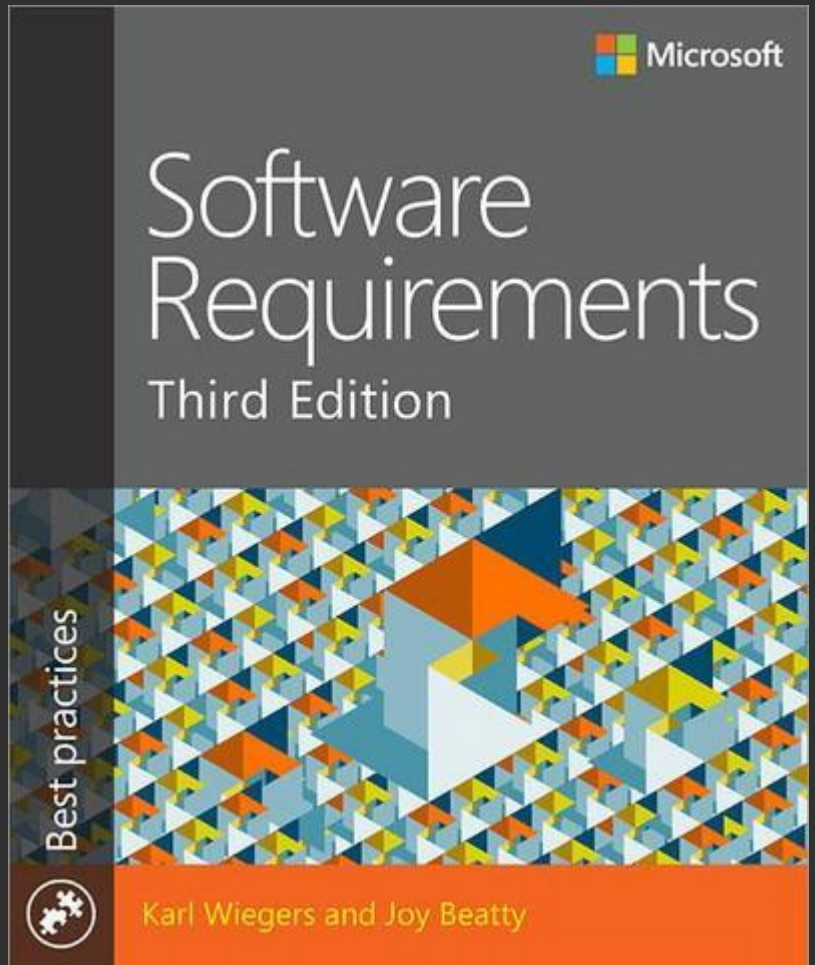
Atividades e resultados no ciclo de vida do software



Representação do SDLC, com explicitação de *QA* (#5) e Manutenção (#6). (Também é comum representar Implementação + QA, #4 e #5 fundidos)

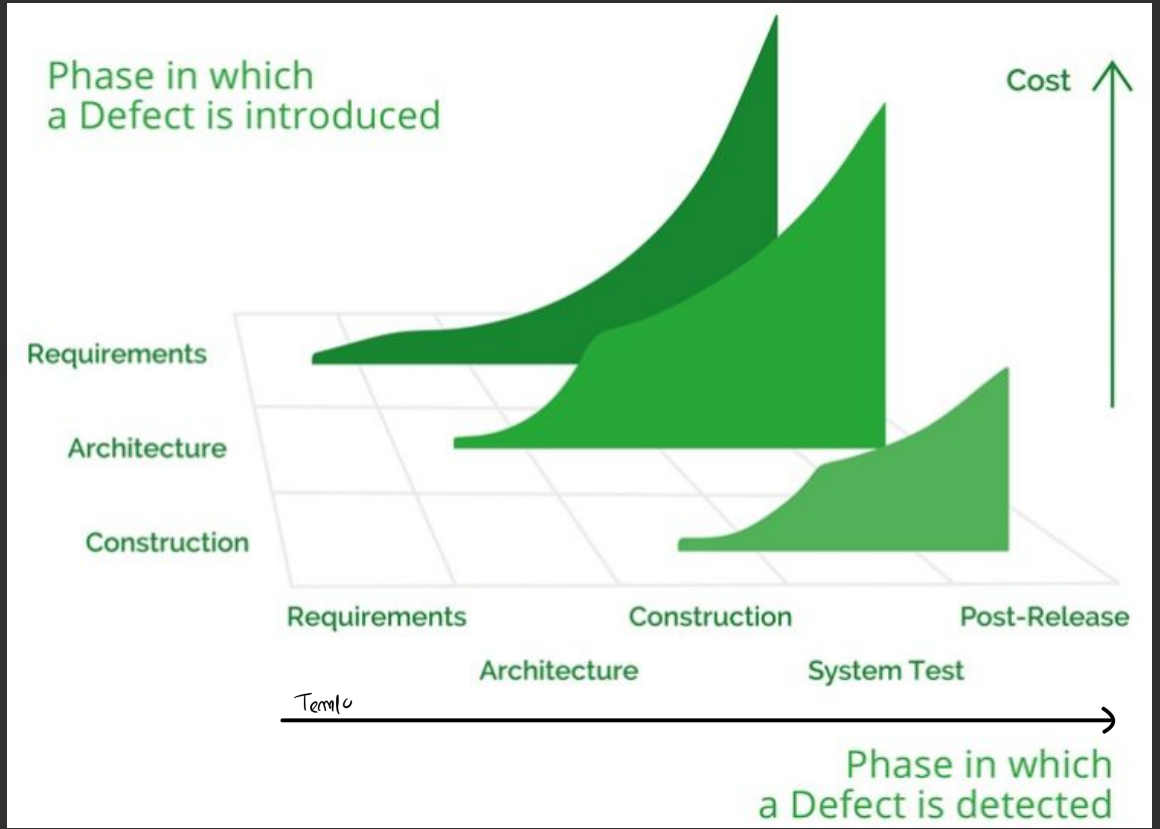
Importância da definição de requisitos

- Definir o âmbito e os limites (da solução/produto)
- Minimizar a necessidade de refazer o trabalho (“*andar para trás...*”) e aumentar a satisfação do cliente
- Servir de base para a conceção (*desenho técnico*), o desenvolvimento e os testes
- Reduzir o risco de insucesso do projeto



O custo do "erro"

- Tendências empíricas...
  - Quanto mais tarde for descoberto, maior é o impacto
  - Definir bem os requisitos



Tipos de requisitos

- Requisitos do utilizador
  - Declarações dos serviços e restrições do sistema na perspetiva do utilizador
- Requisitos funcionais
  - Comportamentos ou funcionalidades específicas do sistema
- Requisitos não funcionais (atributos de qualidade)
  - Desempenho, segurança, facilidade de utilização, fiabilidade, etc.
- Restrições
  - Limitações regulamentares, ambientais ou tecnológicas

Tipos de requisitos

Requisitos funcionais

Funcional: refere-se a comportamento um processo computacional ou tratamento de dados (*as coisas/tarefas que o sistema é capaz de realizar*)

E.g.:

- O sistema deve permitir pesquisar os pacientes pelo nome, usando pelo menos duas palavras (que devem ocorrer em sequência, em qualquer parte do nome).

⇓

Aspetos Funcionais

Requisitos não funcionais

Não-funcional: diz respeito a uma qualidade ou propriedade do sistema (*quão bem deve o sistema fazer a operação*

E.g.:

- As pesquisas de utentes, por nome, retornam os resultados em <1 seg.

⇓

Aspetos de Qualidade

Os atributos de qualidade são necessários na definição do produto

As mesmas capacidades funcionais, mas atributos de qualidade diferentes

Exemplo parcial

Categorias comuns:
- Usabilidade
- Desempenho
- Segurança
- Compatibilidade
- ...

Nonfunctional Requirements

1. Operational Requirements
1.1. The system will operate in Windows environment.
1.2. The system should be able to connect to printers wirelessly.
1.3. The system should automatically back up at the end of each day.

2. Performance Requirements
2.1. The system will store a new appointment in 2 seconds or less.
2.2. The system will retrieve the daily appointment schedule in 2 seconds or less.

3. Security Requirements
3.1. Only doctors can set their availability.
3.2. Only a manager can produce a schedule.

4. Cultural and Political Requirements
4.1. No special cultural and political requirements are anticipated.

Functional Requirements

1. Manage Appointments
1.1. Patient makes new appointment.
1.2. Patient changes appointment.
1.3. Patient cancels appointment.

2. Produce Schedule
2.1. Office Manager checks daily schedule.
2.2. Office Manager prints daily schedule.

3. Record Doctor Availability
3.1. Doctor updates schedule

Agrupamento do RF depende do problema específico

Adapted from: Dennis et al, "*Systems Analysis and Design: An Object Oriented Approach with UML*", 5th ed.

O que pode estar envolvido no "desempenho" (atributo de qualidade)?

TABLE 14-2 Some aspects of performance

Performance dimension	Example
Response time	Number of seconds to display a webpage
Throughput	Credit card transactions processed per second
Data capacity	Maximum number of records stored in a database
Dynamic capacity	Maximum number of concurrent users of a social media website
Predictability in real-time systems	Hard timing requirements for an airplane's flight-control system
Latency	Time delays in music recording and production software
Behavior in degraded modes or overloaded conditions	A natural disaster leads to a massive number of emergency telephone system calls

Credit: Wiegers '13

FURPS: categorias "clássicas“ para os requisitos

Functionality	• avaliado através da apreciação do conjunto de capacidades funcionais do programa.
Usability	• avaliados considerando factores humanos, estética geral, consistência, e documentação.
Reliability	• frequência e gravidade das falhas, da exactidão dos resultados de saída, do tempo médio até à falha (MTTF), da capacidade de recuperação das falhas
Performance	• medida utilizando velocidade de processamento, tempo de resposta, consumo de recursos, rendimento e eficiência.
Supportability	• combina capacidade de extensão, capacidade de **adaptação e de manutenção** e, além disso, compatibilidade, facilidade de configuração, facilidade de instalação e facilidade de teste.

Requirements
STRQ1: Want to be able to transfer funds from other accounts (not necessarily held with this firm) to a trading account.
STRQ2: State and federal regulations require monthly reports of account activity. Refer to specification RUFS-1234 for details of the information required.
STRQ3: The system should allow the use of any browser. S
STRQ4: Customers want to manage their retirement funds. F
STRQ5: Must be able to upgrade the system without taking it offline. P
STRQ6: The system should allow traders to trade in multiple markets across the world. P
STRQ7: Must be able to provide convenient answers to customer's most common questions. U
STRQ8: The system must provide a secure environment that prohibits fraudulent access.
STRQ9: Need a way to train customers in the use of the system quickly and conveniently.
STRQ10: The system must operate on hardware that falls under the company's current maintenance contracts.
STRQ11: Need to be able to maintain the system with our current IT hardware and skills. Refer to enterprise architecture document EA-1234 for details.
STRQ12: Need account activity statements for tax reporting.
STRQ13: The system must provide all the basic capabilities of a normal stock broking firm.
STRQ14: Need to be able to perform research on any given stock.
STRQ15: The system must allow traders to obtain up-to-date news and alerts on nominated stock.
STRQ16: The system must provide current and historical information on Trading Acounts. Such as number of shares held, current price, total Trading Account value
STRQ17: The system shall provide the following types of trades: Market Trades (buy and sell), Limit Trades (buy and sell), and transfers between mutual funds.

Software designers tend to focus on the problem to be solved. Just don't forget that the FURPS attributes are always part of the problem. They must be considered.

A importância relativa dos diferentes atributos de qualidade depende do projeto

Loja on-line muitas sessões em simultâneo, transações seguras,...

App Dialer (telefonar) uso intuitivo, comunicação clara do estado da chamada

Homebanking segurança, disponibilidade,...

Explorar:

- Wiegers, chap. 14

Eng.a de requisitos como um processo

- Levantamento (*Elicitation*)
  - Recolha de requisitos junto dos intervenientes.
- Análise
  - Refinar, priorizar e esclarecer os requisitos.
- Especificação
  - Documentar e organizar os requisitos de forma estruturada.
- Validação
  - Assegurar que os requisitos estão completos e alinhados com as necessidades dos *stakeholders*.
- Gestão dos requisitos
  - Gerir as alterações e manter a integridade dos requisitos.

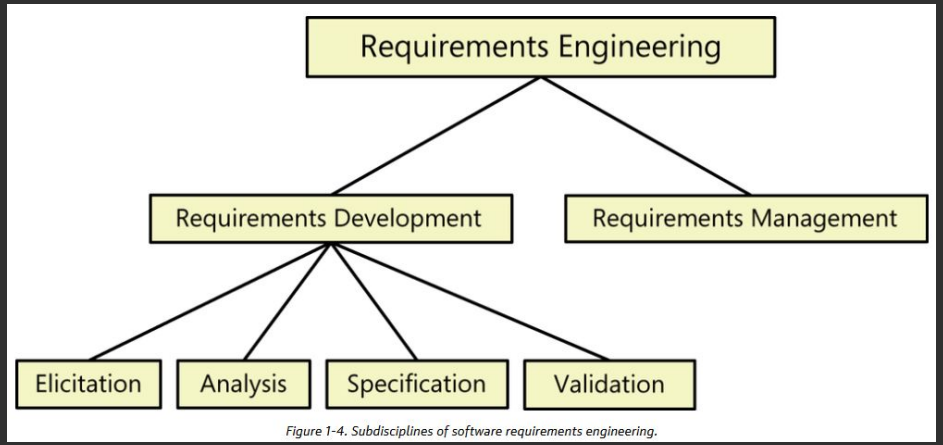


Figure 1-4. Subdisciplines of software requirements engineering.

Quais são as actividades de levantamento dos requisitos?

O núcleo do desenvolvimento de requisitos é a sua identificação

o processo de levantamento das necessidades e restrições dos vários intervenientes para um sistema de software.

Não é apenas uma questão de transcrever exatamente o que os utilizadores dizem...

Elicitação é um processo colaborativo e analítico que inclui actividades para recolher, descobrir, extrair e definir requisitos.

A elicitação é utilizada para descobrir requisitos focados no negócio + focados no utilizador + funcionais + não-funcionais.

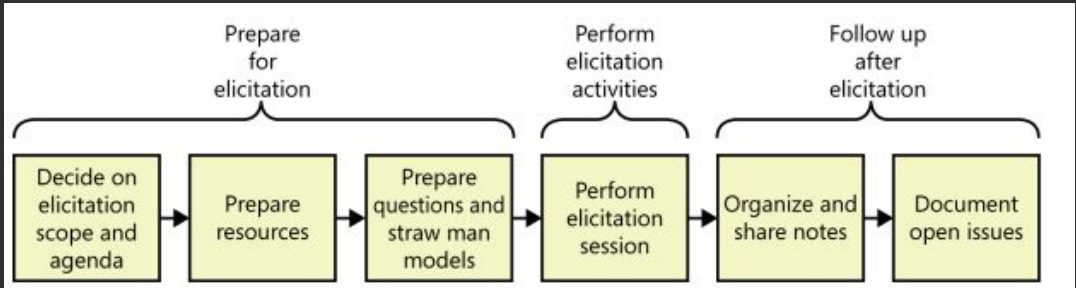
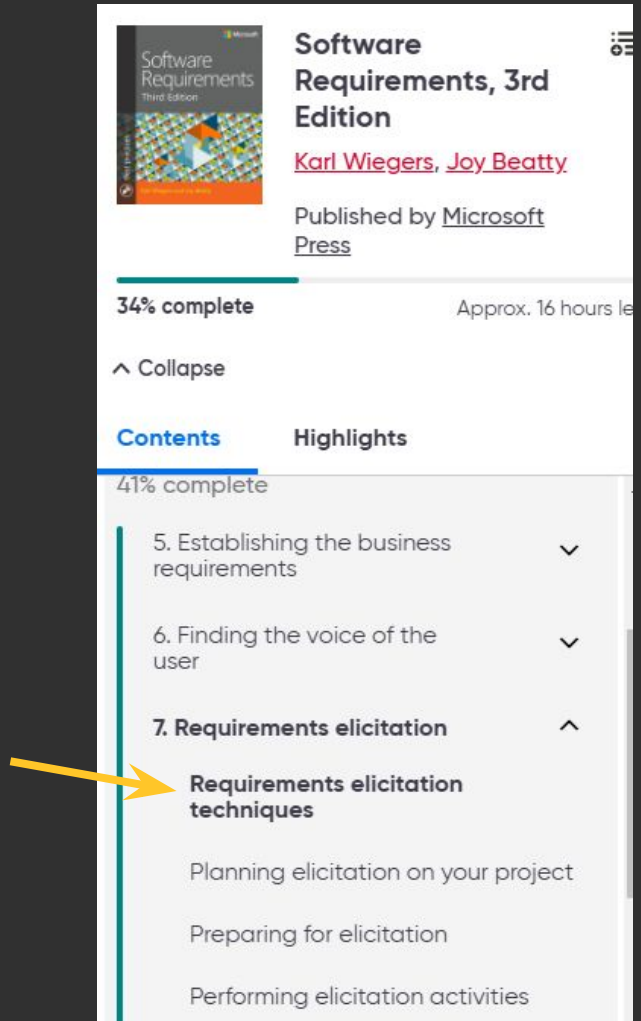


FIGURE 7-2 Activities for a single requirements elicitation session.

Ténicas comuns de levantamento de requisitos

- Entrevistas
- Inquéritos e questionários
- Workshops de desenho colaborativo (JAD) / sessões de brainstorming
- Observação e estudos etnográficos (imersão no ambiente de trabalho)
- Análise de documentos
- *Focus groups*



OpenUP as práticas de requisitos

OpenUP practices

OpenUP > Practices > Technical Practices > Shared Vision > Requirements Gathering Techniques

Detalhes disponíveis na documentação do OpenUp

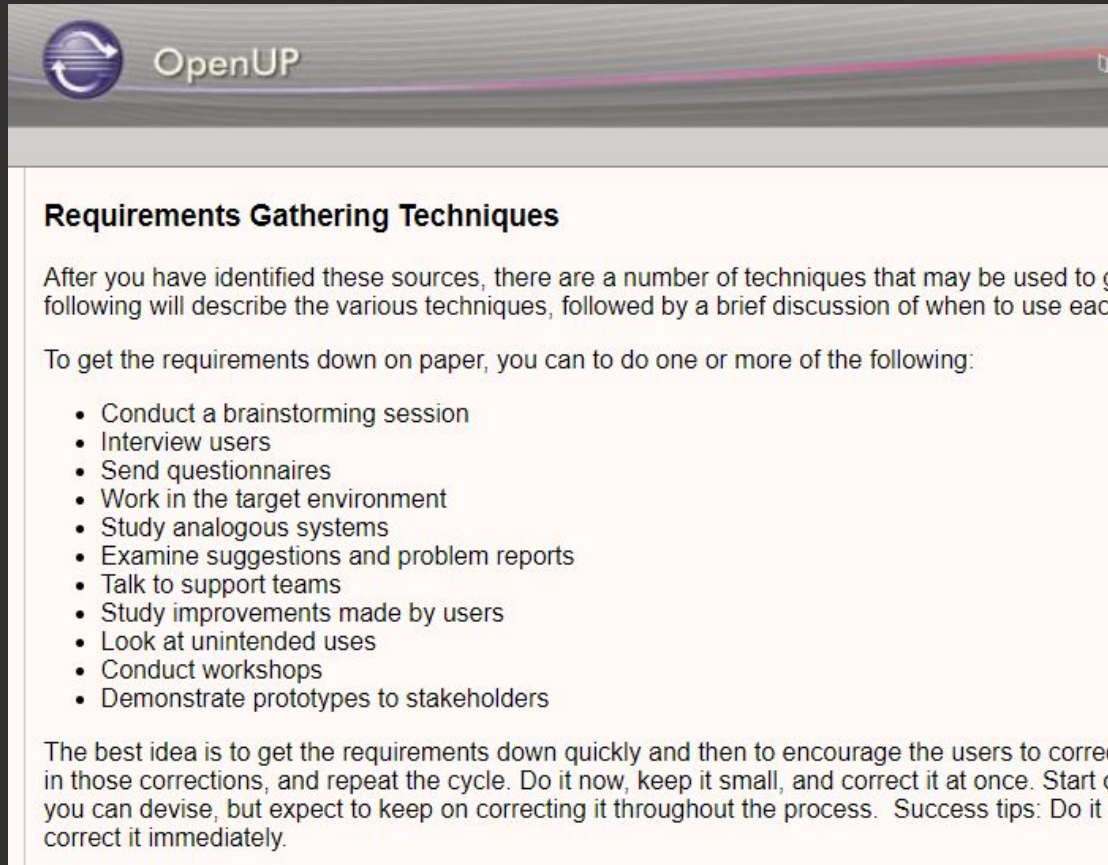


Figure 7-3 suggests the elicitation techniques that are most likely to be useful for various types of projects. Select the row or rows that represent characteristics of your project and read to the right to see which elicitation techniques are most likely to be helpful (marked with an X). For instance, if you're developing a new application, you're likely to get the best results with a combination of stakeholder interviews, workshops, and system interface analysis. Most projects can make use of interviews and workshops. Focus groups are more appropriate than workshops for mass-market software because you have a large external user base but limited access to representatives. These suggestions for elicitation techniques are just that—suggestions. For instance, you might conclude that you do want to apply user interface analysis on mass-market software projects.

	Interviews	Workshops	Focus groups	Observations	Questionnaires	System interface analysis	User interface analysis	Document analysis
Mass-market software	X		X		X			
Internal corporate software	X	X	X	X		X		X
Replacing existing system	X	X		X		X	X	X
Enhancing existing system	X	X				X	X	X
New application	X	X				X		
Packaged software implementation	X	X		X		X		X
Embedded systems	X	X				X		X
Geographically distributed stakeholders	X	X			X			

FIGURE 7-3 Suggested elicitation techniques by project characteristic.

Análise de requisitos

- Definição de prioridades
  - Determinar as caraterísticas de valor elevado versus as caraterísticas facultativas
- Negociação
  - Resolver conflitos entre *stakeholders*
- Modelação
  - Utilizar diagramas e ferramentas (por exemplo, diagramas da UML)
- Ponderação da viabilidade
  - Avaliar as restrições técnicas, económicas e organizacionais

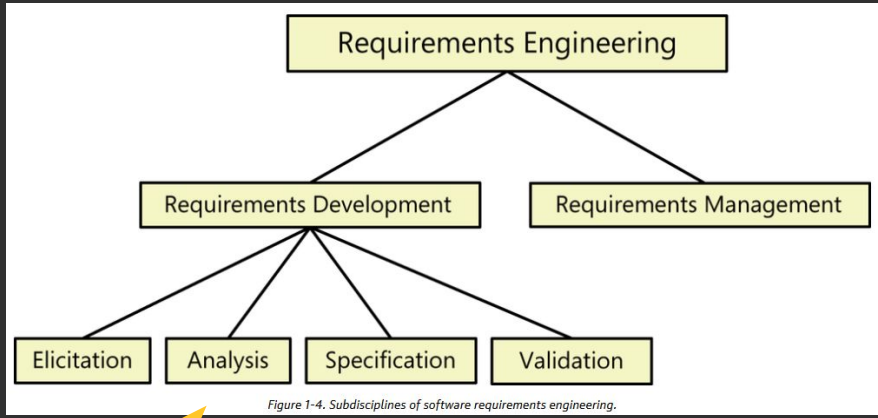
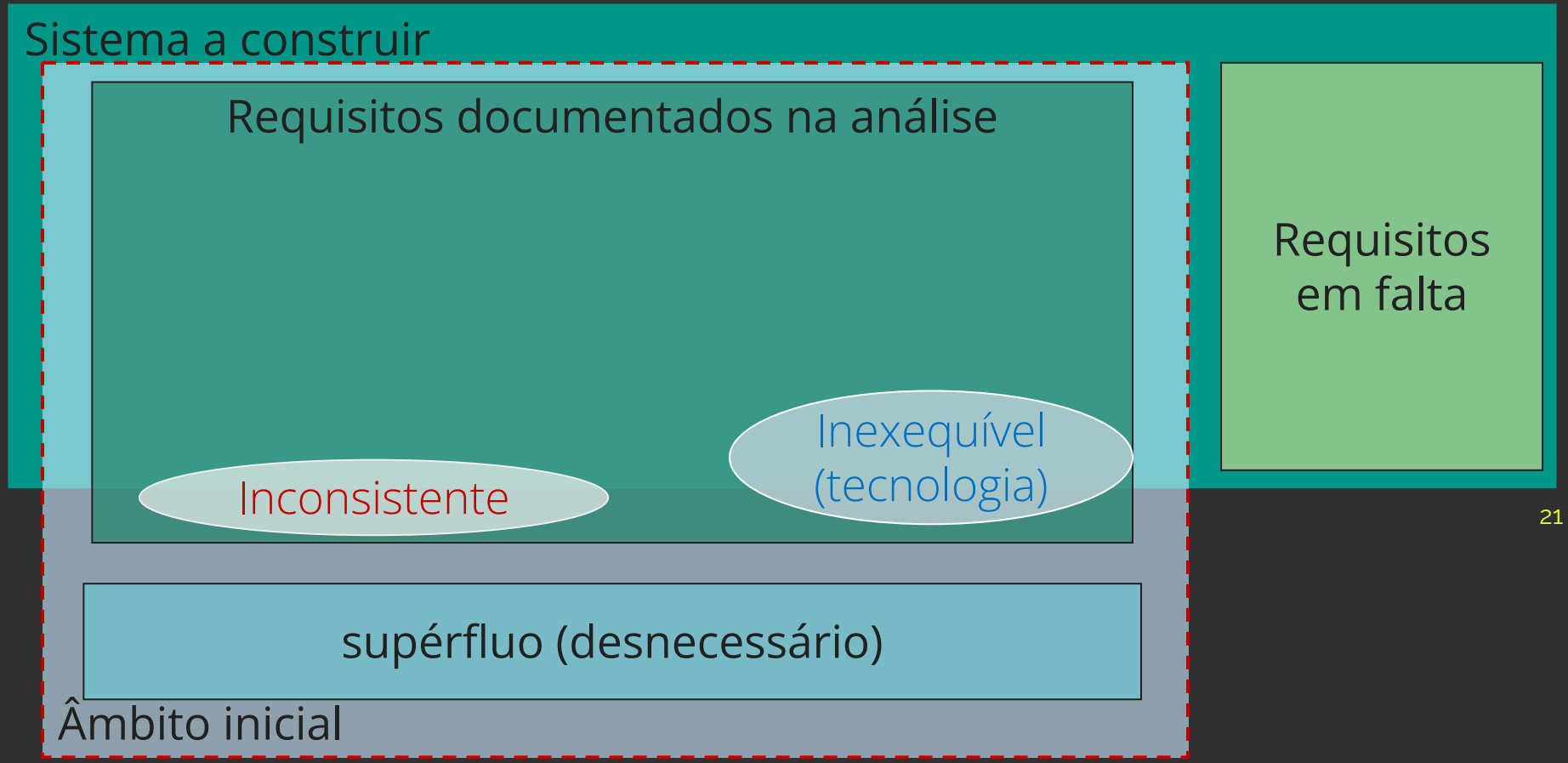


Figure 1-4. Subdisciplines of software requirements engineering.

Seleção dos requisitos



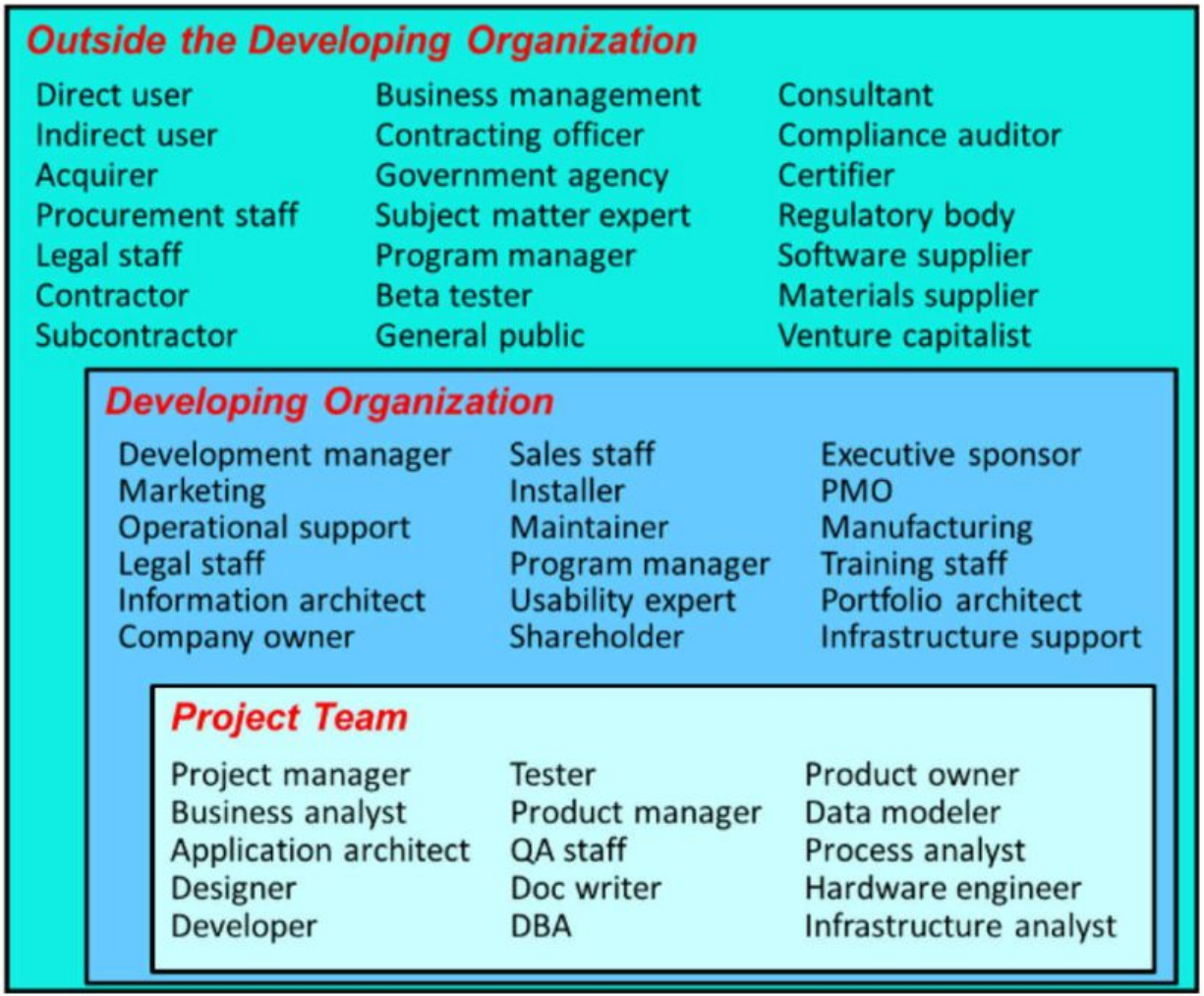


Figure 1. Some possible software project stakeholders.

Especificação de requisitos

- Documentação estruturada e ativamente mantida
- Aplicar uma estratégia coerente
- Abordagens clássicas “product centric”
  - Listas de funções
- Abordagens “user centric”
  - casos de utilização
  - *user stories*

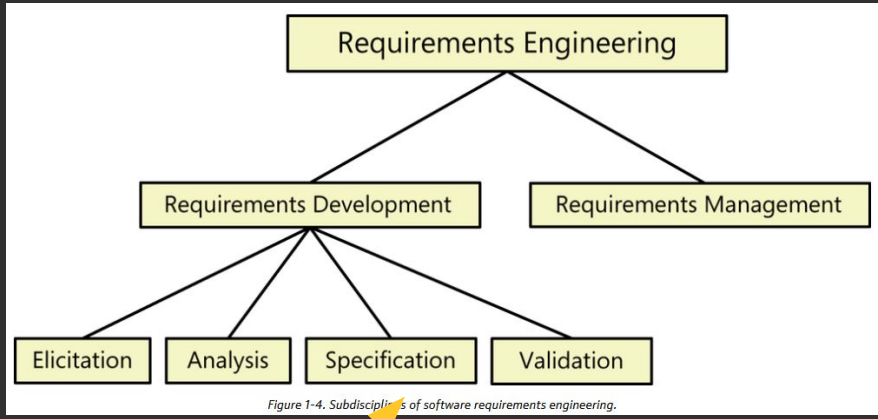


Figure 1-4. Subdisciplines of software requirements engineering.

Requisitos elencados como listas de funções “o sistema deve...”

#	Requisito
RF.1	O sistema deve permitir a um profissional criar um novo pedido de adesão , em auto-serviço, na web.
RF.2	O sistema deve enviar credenciais temporárias para os pedidos de adesão e enviá-las, por email, aos solicitantes, até 10min depois do registo.
RF.3	O sistema deve permitir a pesquisa de cheques-dentista (emitidos) por número de utente do SNS.
RNF.1	As pesquisas de cheques-dentista têm de retornar resultados em <5 segundos ou um evento de tempo expirado.
...	

"The system shall be 100% reliable and 100% available".

"The system shall have a minimum response to a query of 1 second irrespective of system load".

AVL-1. The system shall be at least 95 percent available on weekdays between 6:00 A.M. and midnight Eastern Time, and at least 99 percent available on weekdays between 3:00 P.M. and 5:00 P.M. Eastern Time.

IOP-1. The Chemical Tracking System shall be able to import any valid chemical structure from the ChemDraw (version 13.0 or earlier) and MarvinSketch (version 5.0 or earlier) tools.

PER-1. Authorization of an ATM withdrawal request shall take no more than 2.0 seconds.

PER-2. The anti-lock braking system speed sensors shall report wheel speeds every 2 milliseconds with a variation not to exceed 0.1 millisecond.

PER-3. Webpages shall fully download in an average of 3 seconds or less over a 30 megabits/second Internet connection.

PER-4. At least 98 percent of the time, the trading system shall update the transaction status display within 1 second after the completion of each trade.

S. M. A. R. T. □ Specific, Measurable, Attainable, Relevant and time-sensitive (*Específico, Mensurável, Atingível, Relevante e Balizado no tempo*)

Step 5: Specify well-structured quality requirements

Simplistic quality requirements such as "The system shall be user-friendly" or "The system shall be available 24×7" aren't useful. The former is far too subjective and vague; the latter is rarely realistic or necessary. Neither is measurable. Such requirements provide little guidance to developers. So the final step is to craft specific and verifiable requirements from the information that was elicited regarding each quality attribute. When writing quality requirements, keep in mind the useful SMART mnemonic—make them *Specific*, *Measurable*, *Attainable*, *Relevant*, and *Time-sensitive*.

Credit: Wiegers '13

O que são requisitos bem formulados? (ISO-IEEE 29148)

Uma formulação de uma caraterística que:

- tem de ser satisfeita ou detida por um sistema para resolver um problema ou para atingir um objectivo de alguma "parte interessada"
- pode ser verificada,
- é qualificado por condições mensuráveis e limitado por restrições,
- e define o comportamento ou a capacidade do sistema quando usado, mas não uma capacidade do utilizador ou operador.

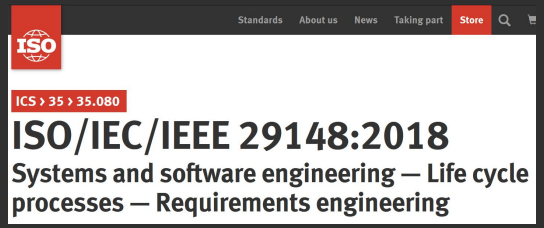
Elementos de estilo

Um requisito é uma declaração que expressa uma necessidade e as limitações e condições que lhe estão associadas.

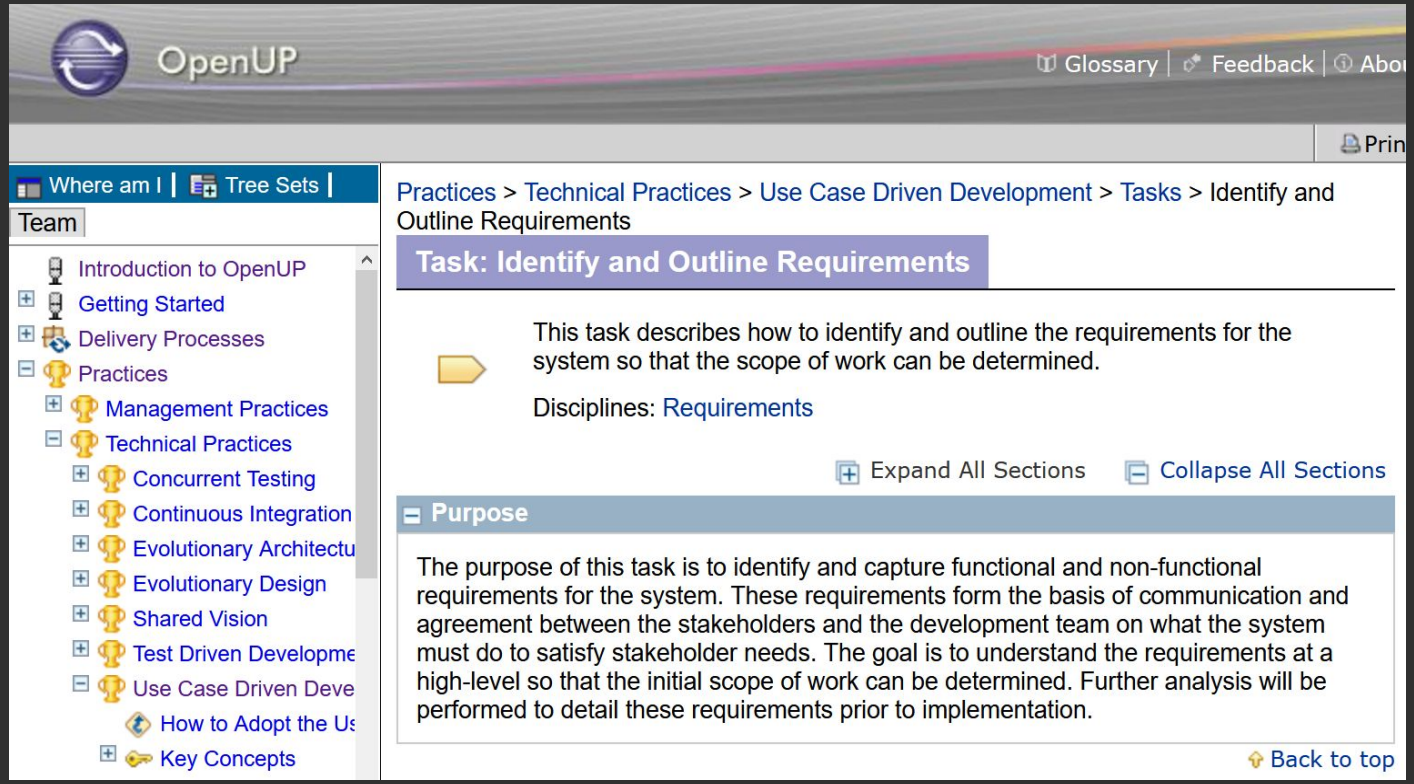
Se expressa em linguagem natural, a declaração deve incluir um sujeito, um verbo e um complemento. Um requisito deve indicar o sujeito do requisito (por exemplo, o sistema, o software, etc.) e o que deve ser feito (por exemplo, operar a um nível de potência).

E.g: O Sistema de Facturação [Sujeito], deve exibir as facturas pendentes do cliente [Acção] por ordem crescente [Complemento] em que as facturas devem ser pagas.

.



OpenUP: como identificar e descrever os requisitos, para que o âmbito fique determinado



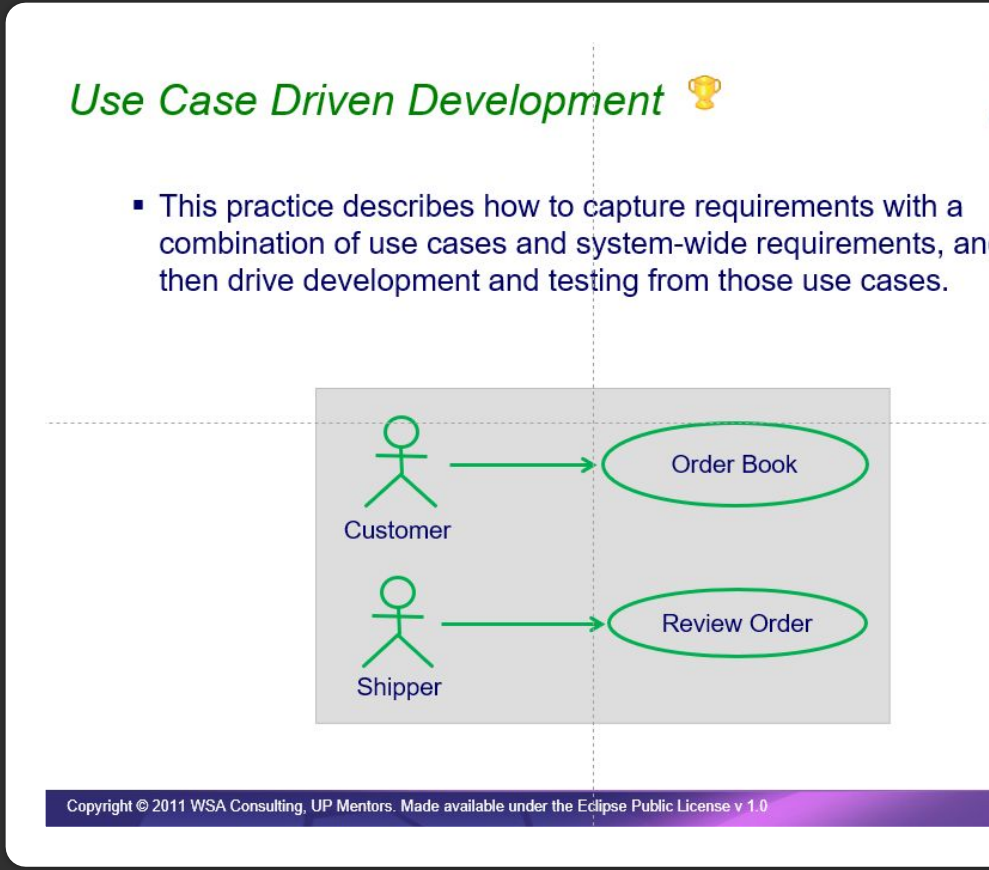
Prática recomendada no OpenUP

Casos de utilização

Engloba diálogos com o sistema que produzem um resultado com valor para algum ator em particular.

Implica:

- **Foco no utilizador** do sistema e nos episódios de uso
- Foco na compreensão daquilo que os atores consideram um **resultado relevante** (motivações para usar um sistema problemas que querem resolver)

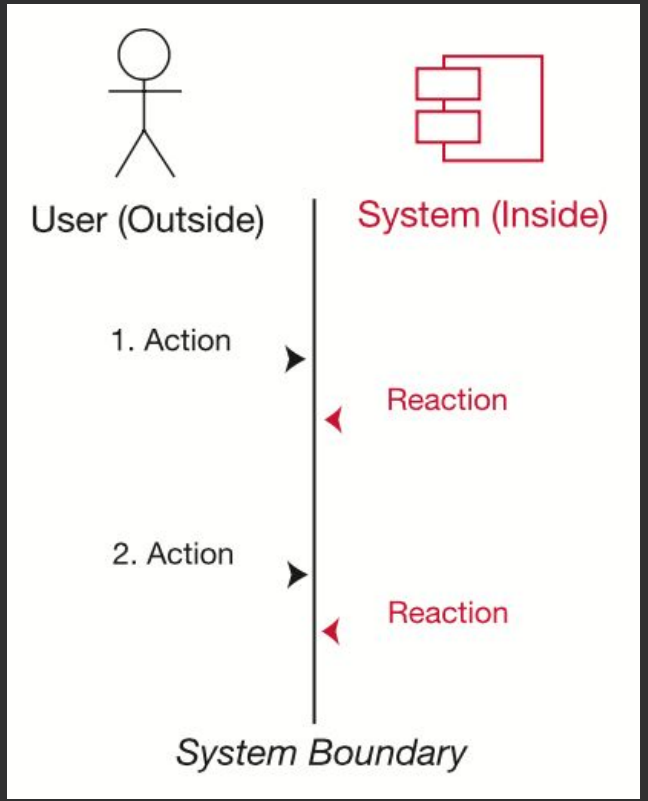


O CaU descreve **um diálogo entre o ator e o sistema**

Narrativas para contar como é que “alguém” usa um sistema

CaU: Comprar produtos (supermercado)

1. Um Cliente chega a uma caixa com artigos para comprar.
2. O Operador passa cada artigo no leitor de código de barras para registo.
3. O sistema apresenta o total provisório e a lista de artigos incluídos.
4. O Operador termina a venda e indica o tipo de pagamento.
5. O cliente introduz a informação de pagamento.
6. O sistema valida o pagamento, atualiza o stock e imprime o recibo.
7. O cliente leva o recibo e os artigos.



Use case:	Brief description:
Create new assigment	The Teaching Staff creates a new Activity of type Assignment, directly inserting it in the page layout. The assignment must define a title and a time period for submissions and can be configured to work with individual or group submissions. The assignment is listed in the student view and on the specified date (or immediately, if none is given) accepts submissions from registered students.

Use case:	Add new assigment
Brief description:	The Faculty creates assignments for students, directly inserting it in the course page. The assignment defines a time period for submissions and can be configured to work with individual or group submissions. The assignment is listed in the student view and on the specified date (or immediately, if none is given) accepts submissions from students.
Basic flow:	1. Log-in using corporate IdP. 2. Select desired course. 3. Turn editing mode on. 4. Add Assignment activity in the page layout. 5. Configure Assignment activity. 6. Commit changes.
Alternative flows:	Step 1: IdP unavailable. Step 4/5: Instead of a new, empty assignment, the user may reuse an existing one.
Open issues:	Step 3/4. The course is closed. Are changes allowed to past courses? Step 5. The browser does not accept the rich text editor. Default to plain text?

Exemplo...

Elementos de um caso de utilização

Nome do caso de utilização - Título claro e descritivo

Ator(es) - Papéis que participam no cenário de interação com o sistema

Condições prévias - O que tem de ser verdade antes de o caso de utilização começar

Fluxo de eventos (caminho principal) - Descrição passo a passo das interações

Fluxos alternativos/excepções - Variações e tratamento de erros

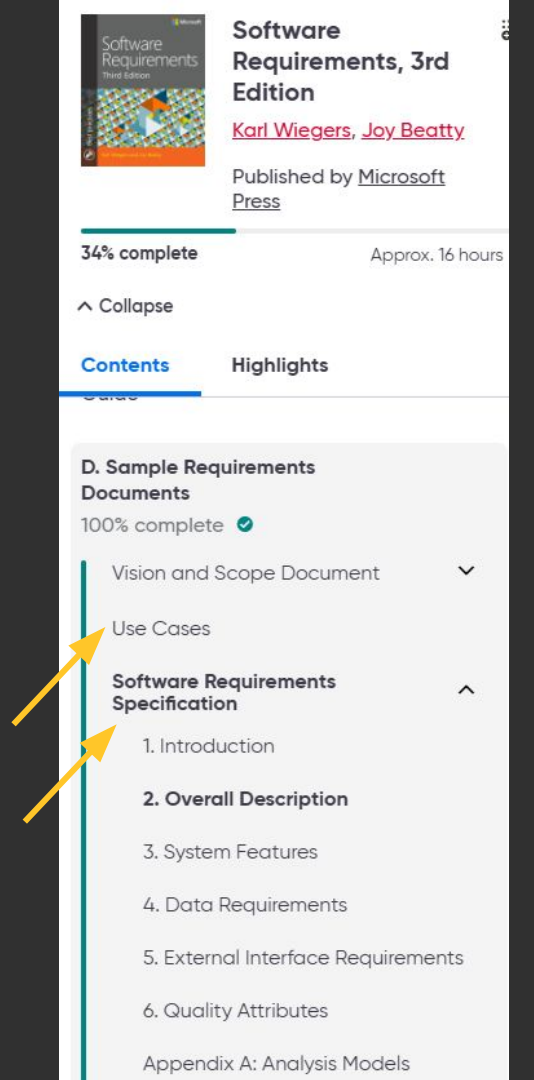
Pós-condições - Estado final após a conclusão do caso de utilização

Pressupostos - Condições assumidas mas não garantidas

Exemplos de especificações

SRS: Software Requirements Specification

- Exemplo 1: Use cases
- Exemplo 2: SRS (Wiegers)
- Exemplo 3: SRS (clássico)



Conclusões...

Boas práticas

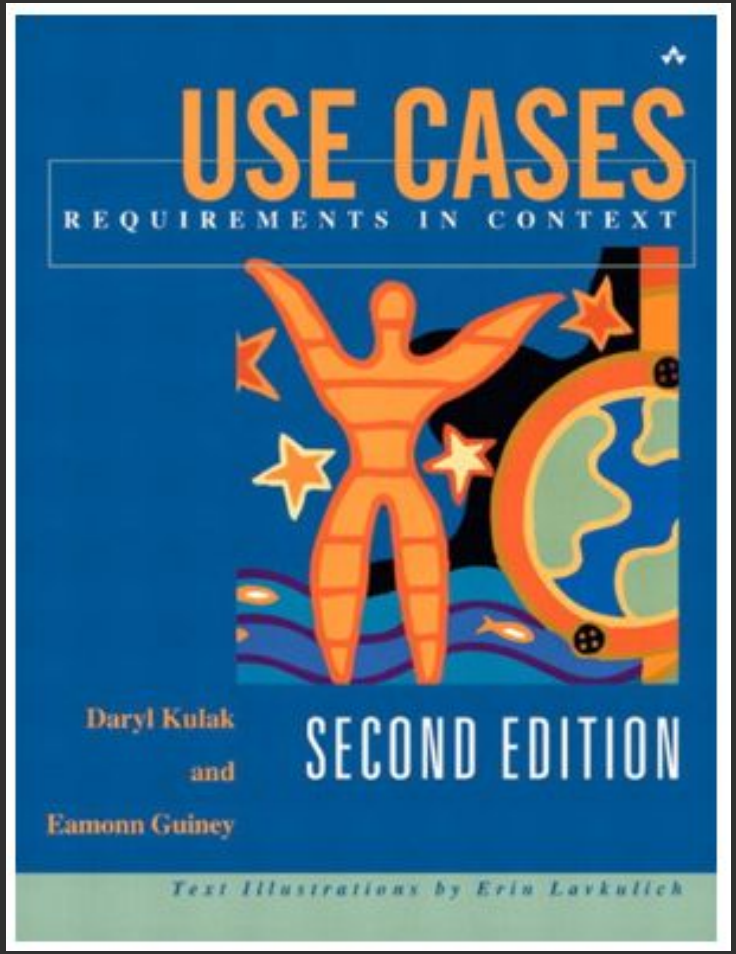
Envolver as partes interessadas desde o início e com frequência

Equilibrar o pormenor com a lucidez

Utilizar modelos e ferramentas coerentes

Validar e rever os requisitos de forma colaborativa

Tirar partido dos casos de utilização para uma comunicação mais clara

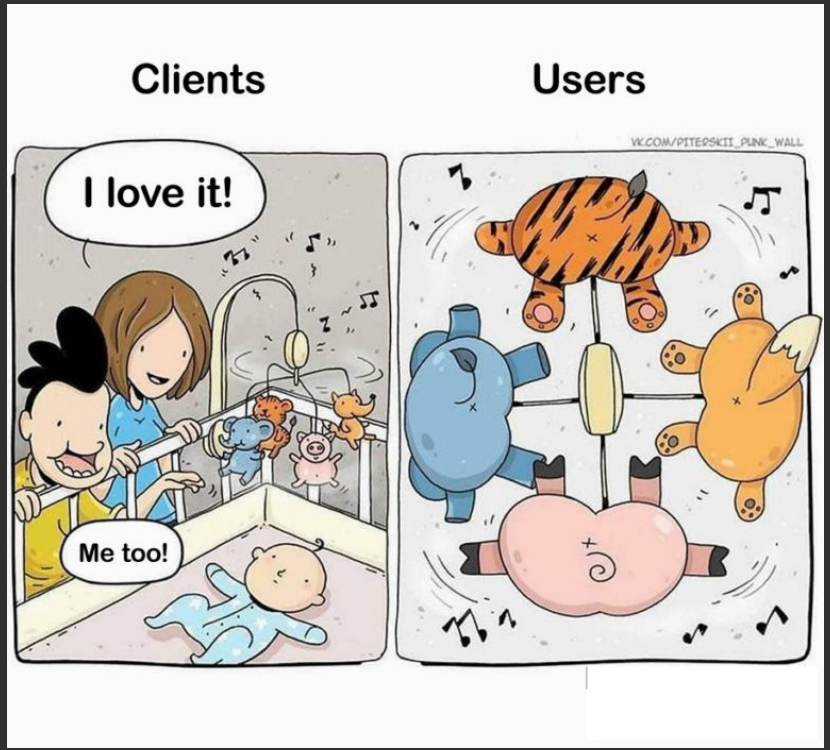


Compromisso...

Valor para o negócio...

Utilizadores...

Viabilidade...



https://www.designershumor.com/2019/05/21/ux-is-important/

And second, those of us in the software domain tend to be enamored with technical and process solutions to our challenges. We sometimes fail to appreciate that requirements elicitation—and much of software and systems project work in general—is primarily a human interaction challenge. No magical new techniques have come along to automate that, although various tools are available to help geographically separated people collaborate effectively.

In: Wiegers, “Software Requirements”