

2. Modelos de análise

práticas de engenharia de requisitos

requisito → declaração de que o sistema deve fazer ou de uma característica que deve ter

funcionais contêm o comportamento pretendido do sistema (função, serviços ou tarefas, por exemplo)
podem ser captados nos casos descritos em diagramas de atividades / sequência
diagramas de comportamento

não funcionais também chamados de atributos de qualidade se não forem levados a cabo o sistema torna-se inconsistente e de baixa qualidade
características que sistema terá de exibir
restrições que terá de respeitar

atributos de qualidade FRAPS classificações	Funcionalidade	capacidades lógicas	ex. sequência
	Usabilidade	capacidades documentais	ex. dar dinheiro (ATM)
	Reliability	disponibilidade (tempo ativo)	
	Performance	tempo de resposta, ligação, recuperação, etc.	
	Support	teste, adaptação, compatibilidade, manutenção	
	classificações dos atributos de qualidade		

avaliação de requisitos SMART	Specific	ter metas específicas
	Mediável	ter métricas objetivas
	Attingível	executível
	Relevante	ter significado prático
	Temporal	meta definida

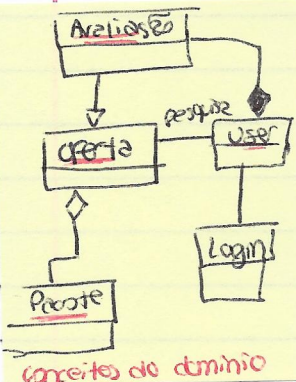
→ identifica requisitos bem e mal formulados

os requisitos passam por um processo de engenharia de software para obterem um sistema novo

engenharia de software | conjunto das atividades e resultados associados que produzem uma peça de software

modelo do domínio

é um mapa por os objetos
mostra os conceitos de um problema "trocar por miúdos"
representado por um diagrama de classe sem operações
representa objetos, conceitos e associações entre eles, atributos
não tem a implementação!



modelação do contexto do problema

classes de análise
MAS

resultado da análise dos requisitos
neutro em relação à implementação
é apenas modelo de dados nem (classes) de programação

construção

1. identificar objetos do problema
2. representar as classes num DC (diagrama de classe)
3. adicionar associações e atributos

decomposição do problema

algorítmica

começamos com um modelo geral que vamos "dividir" em modelos cada vez mais simples, criando uma estrutura em árvore que mostra as relações entre os elementos funcionais da solução

orientada a objetos

em vez de decompor o problema em vários passos, consiste na identificação dos objetos envolvidos e das relações entre eles

- vs
- a algorítmica destaca a ordem dos eventos
 - a orientada enfatiza os agentes que praticam ou são alvo de uma determinada ação

associações entre conceitos

descoberte

na forma [entidade] [verbo]
forms/reports da área do problema ligam info de várias entidades
antesão, tmbo são um elemento de modelação

práticas

associações binárias
devem ter nome para clarificar interpretação
le-se de cima p/ baixo e da esq p/ direita

modelação funcional
com casos de
utilização

identificar casos de utilização	entrevistas desenvolvimento conjunto da aplicação questionários análise dos documentos observação, brainstorming, ... permitem descobrir todos os requisitos e fomentam confiança nos utilizadores
---------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

casos de utilização	<u>propósito</u> objetivo <u>sequência</u> ordem da "narrativa"
---------------------	--------------------------------------------------------------------

UML Case 2.0	diagramas de atividade relaciona casos de utilização de CAU especificam um caso de utilização específico narrativa dos casos de uso especifica cada cenário do CAU
--------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

UC01. Register Client

Fluxo 1. Abrir formulário base
2. Informar das condições de venda
3. ...

Fluxo Alternativa 1 2.1. Caso na DE, mostrar POPD

Fluxo Alternativa 2 2.1. Caso na EVA, mostrar doc y

princípios para a adoção dos casos de utilização

1. manter simplicidade
contar histórias e forma + simples de transmitir conhecimento
2. ver o quadro geral (big picture)
compreender o sistema como um todo é essencial
um diagrama de CAU é uma forma simples de apresentar uma visão geral dos requisitos de um sistema
3. foco no valor
valor é gerado quando o sistema é usado
os CAU focam-se em mostrar o valor que o sistema tem ao distinguir os objetivos distintos para diferentes utilizadores
4. fatias
sistemas complexos devem ser implementados em partes
5. entregas incrementais
cada incremento fornece uma versão demonstrável e utilizável
6. adaptações
princípios são flexíveis à realidade de cada projeto

requisitos no OpenUP	<u>fontes</u>	clientes, utilizadores, administradores, parceiros, analistas da indústria, informação sobre a concorrência
	<u>métodos</u>	brainstorming, entrevistas, questionários, estudo de sistemas análogos, workshops, utilizadores involuntários

modelos estruturais são utilizados para especificar sistemas

diagramas de
classes e
objetos

modelos
estrutural

modelos
de comportamento