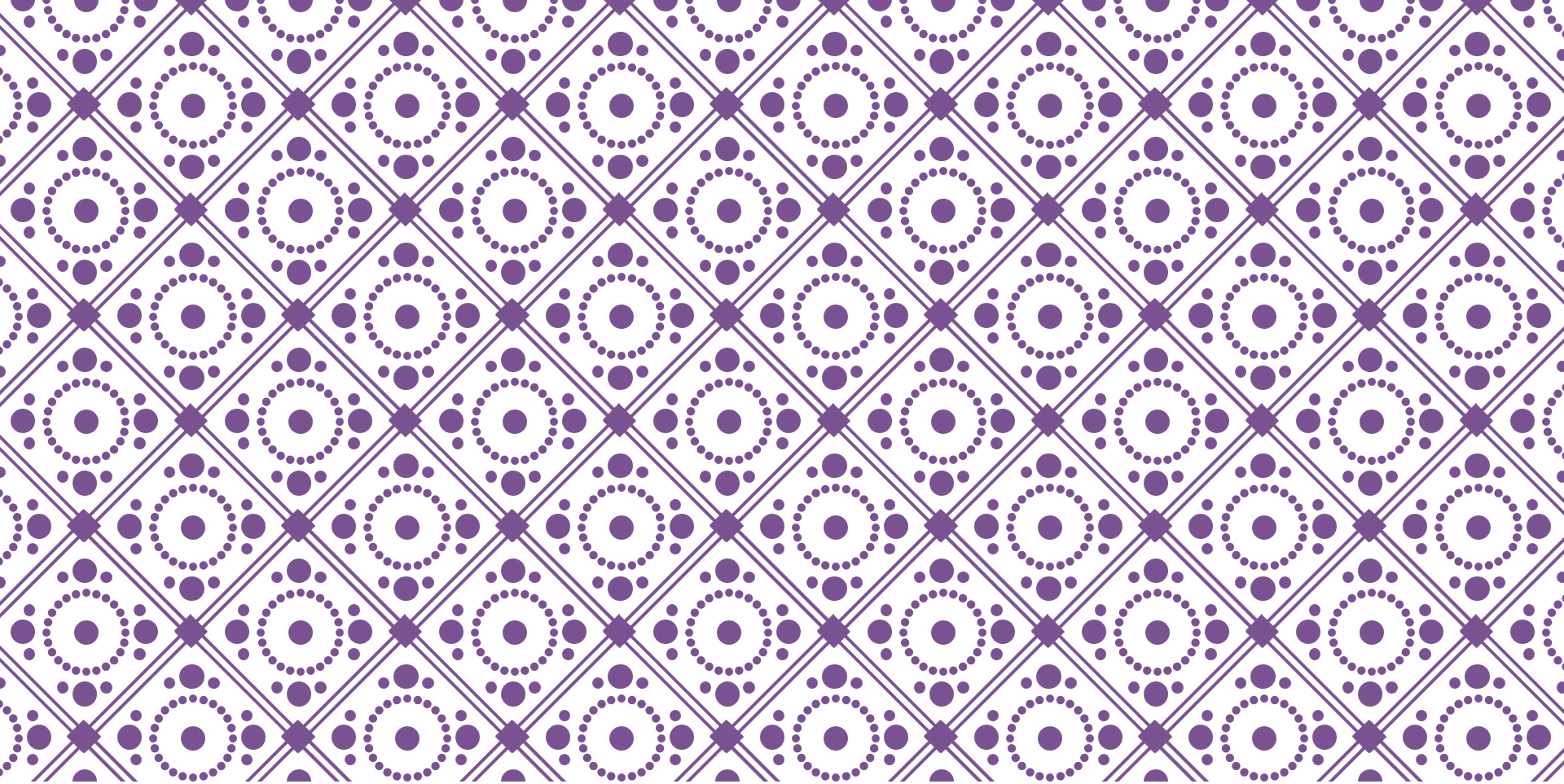
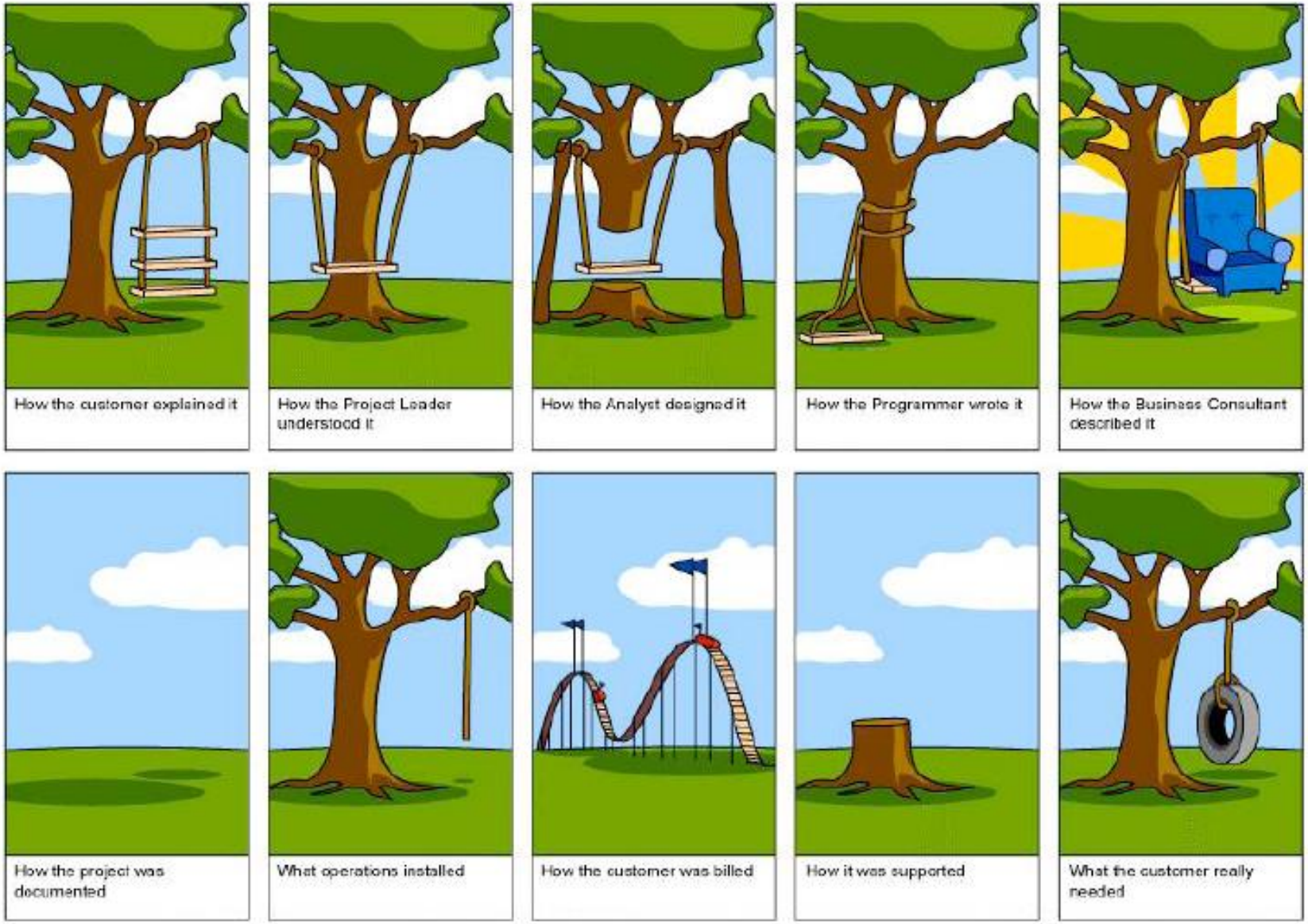


PROJETO DA INTERAÇÃO

Interação Pessoa Máquina

Anabela Gomes


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

PROJETO

Qualquer disciplina de engenharia envolve a aplicação estruturada de técnicas para o desenvolvimento de um produto

A Engenharia de Software fornece um meio de compreender a estrutura e aplicar as técnicas de desenvolvimento de software

O ciclo de vida do software pretende identificar as atividades que ocorrem no desenvolvimento de software

O desenvolvimento de software envolve 2 partes principais

- Utilizador (quem compra (cliente) ≠ quem usa (utilizador))
- Projetista (designer)

PROJETO


BEHOLD!!!
THIS IS A TESTAMENT TO WHAT CAN HAPPEN WHEN YOU LISTEN TO CUSTOMERS.
WE ASKED CUSTOMERS WHAT THEY WANTED THE NEW VERSION TO DO.
SPIT

CICLO DE VIDA DO SOFTWARE

60´s e 70´s – crise de software

- Code now, fix it later...

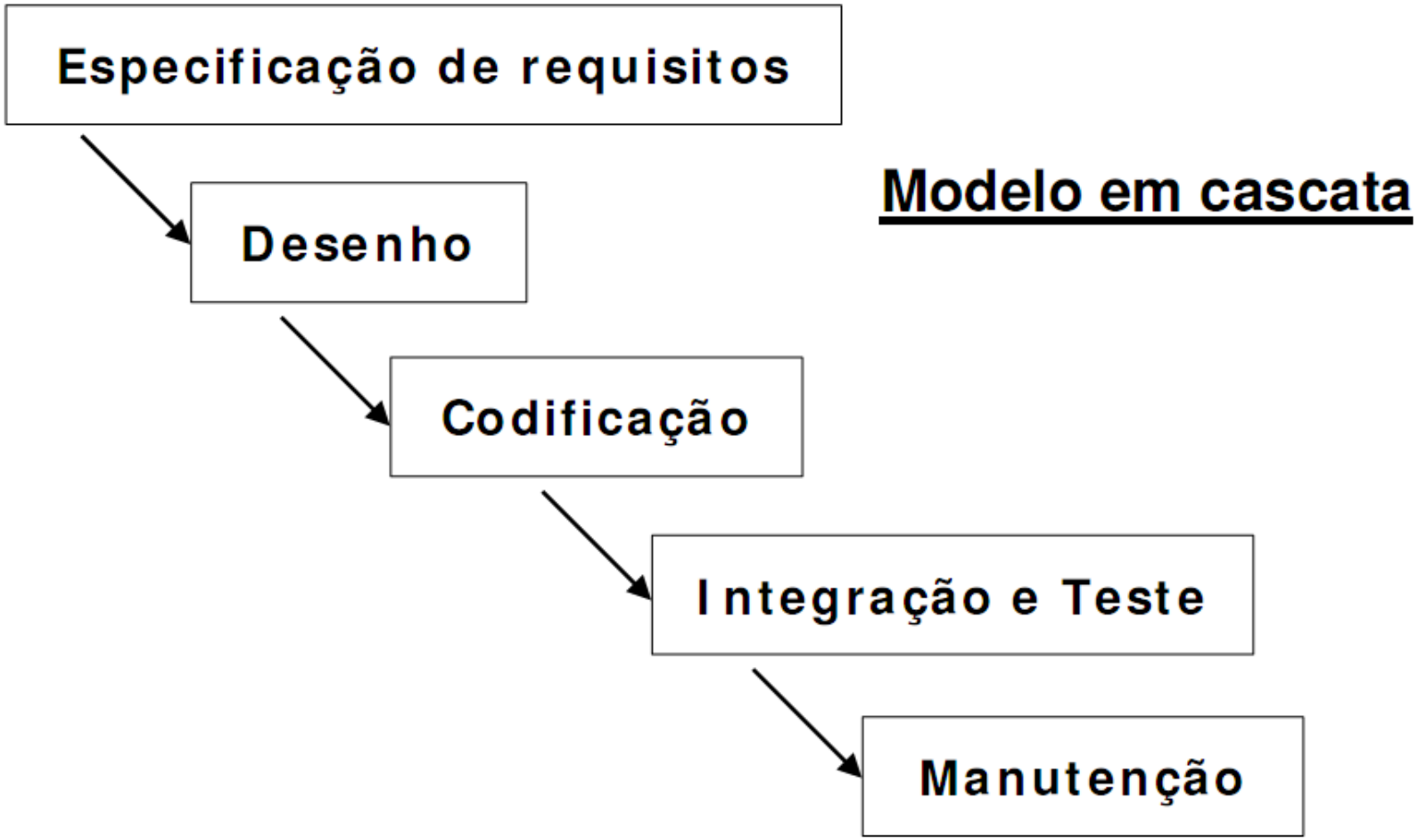
Modelo cascata ("waterfall model")

- "think first and code second"

Desenho iterativo

- Alterações numa atividade influenciam as outras atividades

CICLO DE VIDA DO SOFTWARE



Modelo em cascata

CICLO DE VIDA DO SOFTWARE

Especificação de requisitos

- O projetista e o utilizador tentam determinar **o que** deve ser implementado
- Envolve a recolha de informação acerca do **utilizador** e **ambiente de trabalho** ou domínio, no qual o produto final vai funcionar
- Aspetos do domínio de trabalho incluem
  - funções a desempenhar pelo software
  - pessoas que vai potencialmente afetar
  - relação com produtos existentes que vai atualizar ou substituir
- Deve ser expressa numa linguagem compreendida por ambos os intervenientes no processo

Análise de tarefas

CICLO DE VIDA DO SOFTWARE

Desenho

- Determina **como** o sistema deve ser implementado de modo a cumprir os requisitos especificados
- Decomposição do sistema em componentes
  - decomposição funcional: que componentes fornecem que serviços
  - descrição de interdependências e partilha de recursos entre componentes
- **Requisitos Funcionais:** serviços que o sistema tem que fornecer no domínio de trabalho
- **Requisitos não-Funcionais**: relacionados com o modo de disponibilização dos serviços: eficiência, confiança, *timing,* segurança e características interativas
- Objetivo: produzir descrição suficientemente detalhada de cada componente e do modo de integração dos diversos componentes, de forma a que possam ser implementados numa qualquer linguagem de programação

CICLO DE VIDA DO SOFTWARE

Codificação

- Implementação e teste dos módulos individuais numa qualquer linguagem de programação

Integração e Teste

- Assim que os componentes estejam implementados e testados individualmente, têm que ser integrados
- Testes de aceitação com os clientes para verificar cumprimento dos requisitos
- Testes para certificar comportamento correto e uso aceitável de quaisquer recursos partilhados
- Pode ainda ser necessário certificar o sistema perante alguma autoridade externa (Ex.: aplicações financeiras).

CICLO DE VIDA DO SOFTWARE

Manutenção

- Depois do lançamento do produto, todo o trabalho efetuado no sistema é considerado atividade de manutenção, até que seja produzida uma nova versão completamente redesenhada ou o produto seja retirado de atividade
- Maior fatia do ciclo de vida
- Envolve
  - Correção de erros descobertos depois do lançamento
  - Revisão dos serviços do sistema para satisfazer requisitos não identificados previamente
- Esta atividade dá feedback a todas as outras

CICLO DE VIDA DO SOFTWARE

Validação - desenhado o produto certo?

- Exercício mais subjetivo que de verificação
- Deve demonstrar que dentro de todas as atividades do CV, os requisitos dos clientes são satisfeitos
- No desenho de sistemas interativos, a validação dos requisitos de HCI denomina-se avaliação (evaluation).

Verificação - produto bem desenhado?

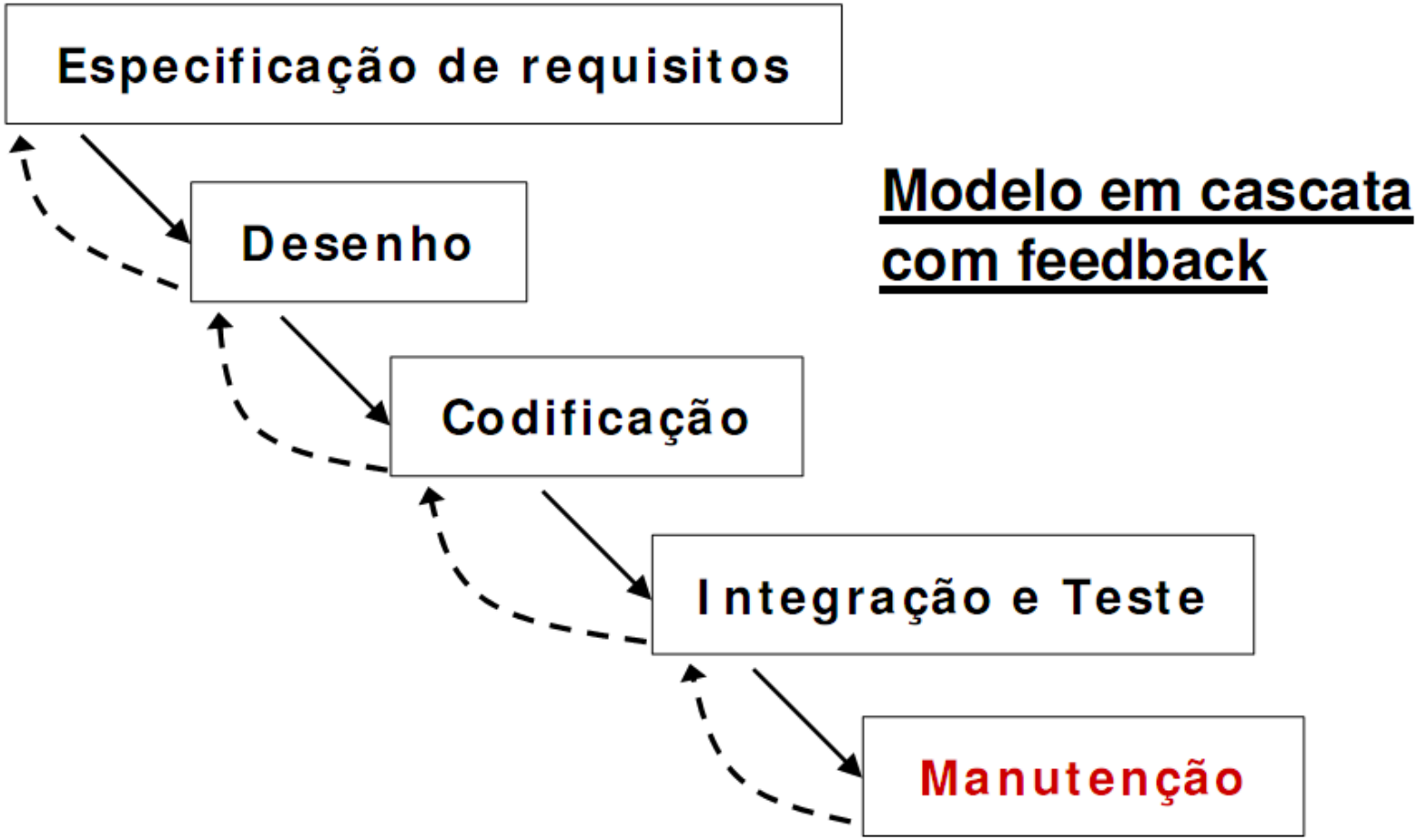
- Ocorre dentro de uma atividade do CV ou quando muito entre duas atividades adjacentes
  - Ex.: no desenho dum componente duma aplicação de vencimentos, o projetista tem de fazer a verificação do algoritmo que determina os ordenados

CICLO DE VIDA DO SOFTWARE

Modelo em cascata não é apropriado para o desenho de interfaces de sistemas interativos

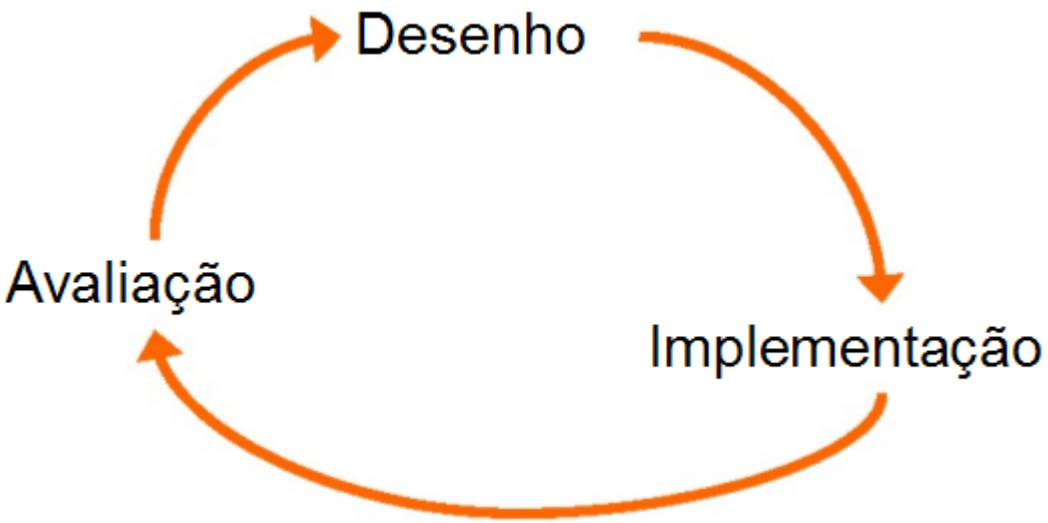
- Os utilizadores apenas intervêm na especificação de requisitos e nos testes
- Deteção tardia de erros resulta em correções demoradas e dispendiosas
- Não suporta processos verdadeiramente iterativos

CICLO DE VIDA DO SOFTWARE



CICLO DE VIDA DO SOFTWARE

Desenho iterativo



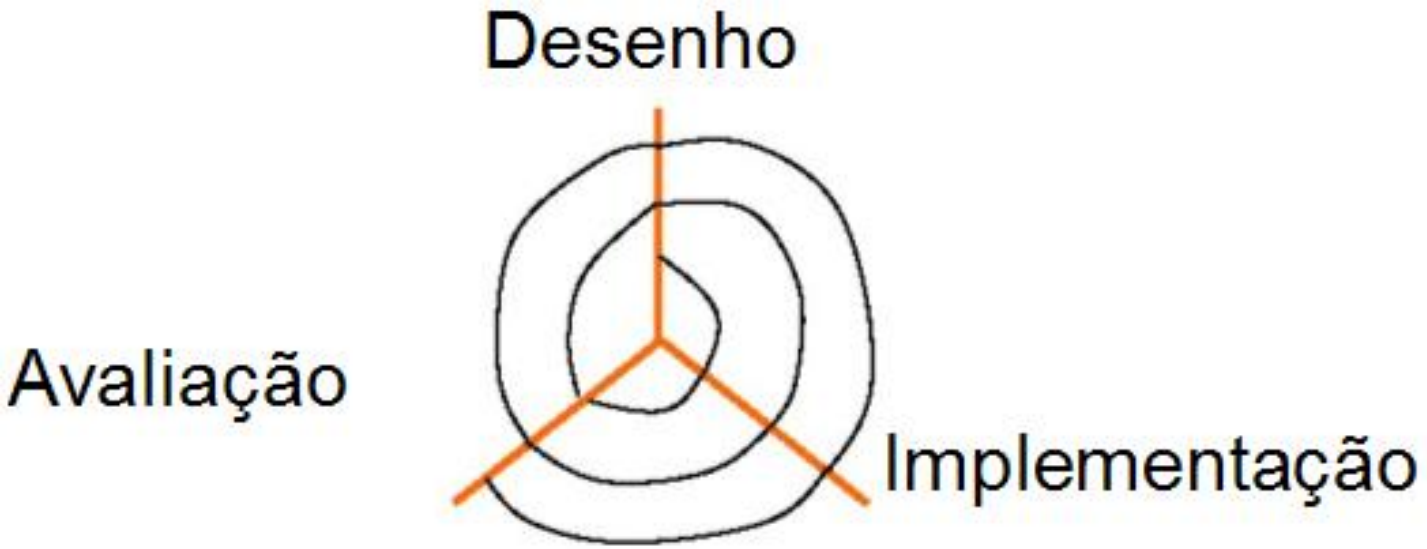
CICLO DE VIDA DO SOFTWARE

Desenho iterativo

- Modo incorreto: segue o modelo em cascata, produz uma má interface e lança o sistema
  - Cada iteração corresponde a uma nova versão
    - Erros detetados na avaliação são corrigidos na versão seguinte
  - O cliente é usado para avaliar a usabilidade do sistema
    - Se não gostarem não compram a versão seguinte.

CICLO DE VIDA DO SOFTWARE

Desenho iterativo -> Modelo em espiral



CICLO DE VIDA DO SOFTWARE

Modelo espiral

- Várias iterações
  - Custo, precisão e correcção aumentam de iteração para iteração
- Primeira iteração pode ser efectuada em papel: baixo custo e ainda bastante longe do sistema real

CICLO DE VIDA DO SOFTWARE

Primeiras iterações usam protótipos baratos

- Desenho paralelo: construção e teste de vários protótipos para explorar múltiplas alternativas

Iterações seguintes (depois de eliminados os maiores riscos) criam-se protótipos mais elaborados

Todos os protótipos são avaliados

- Os utilizadores são envolvidos em todas as iterações

Mais iterações => melhores interfaces

Só as melhores iterações sobrevivem e surgem no mercado

CICLO DE VIDA DO SOFTWARE

O teste de um protótipo (ex.:papel) da interface numa iteração inicial permite detetar muitos problemas a evitar. Nessa fase a alteração do desenho custa apenas mais um "sketch".

Case study

- The 1984 Olympic Message System: a test of behavioral principles of system design by John Gould et al. Communications of ACM, v.30 n.9, 1987

- http://www.youtube.com/watch?v=W6UYpXc4czM

OUTRAS METODOLOGIAS

Behavior-driven development (BDD)

Design-driven development (D3)

Domain-driven design (DDD)

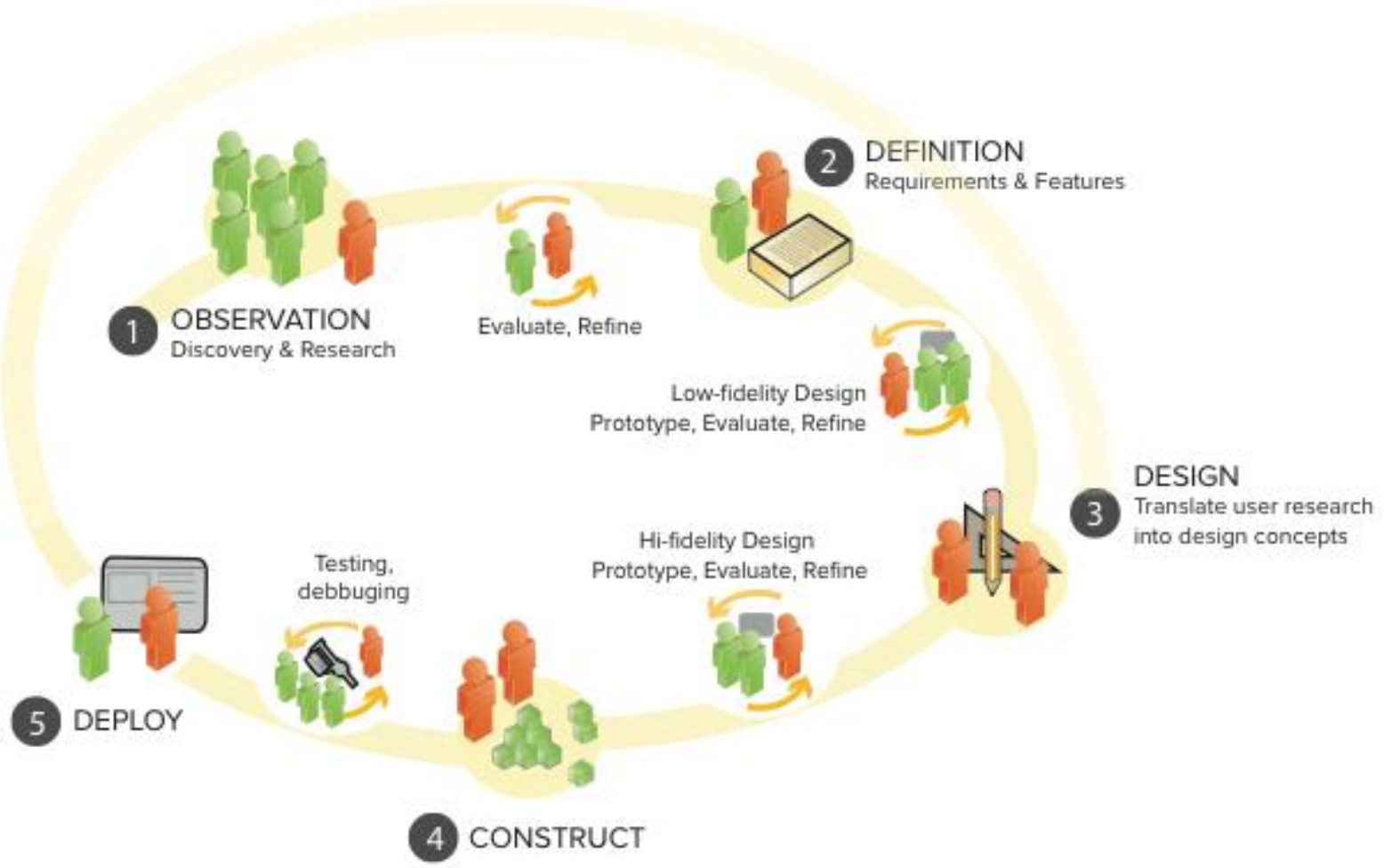
Feature-driven development (FDD)

Test-driven development (TDD)

Value-driven design (VDD)

User-centered design (UCD)

DESENHO CENTRADO NO UTILIZADOR



DESENHO CENTRADO NO UTILIZADOR

Foco nos utilizadores desde uma fase inicial

Testes com utilizadores desde início e contínuo

Desenho iterativo

Colaboração entre projetistas e utilizadores

Utilizadores e projetistas em constante comunicação

O projeto adapta-se e evolui em função das necessidades

CASE STUDY: OMS

Foco nos **utilizadores** e nas **tarefas**: estudo cognitivo e comportamental do utilizador, **desenho participativo**

Medições empíricas: reacções dos **utilizadores** aos **cenários** definidos, simulações e **protótipos**

Desenho iterativo

USER-CENTERED DESIGN

O desenho deve ser baseado nas

- necessidades
- habilidades
- contexto
- trabalho
- tarefas

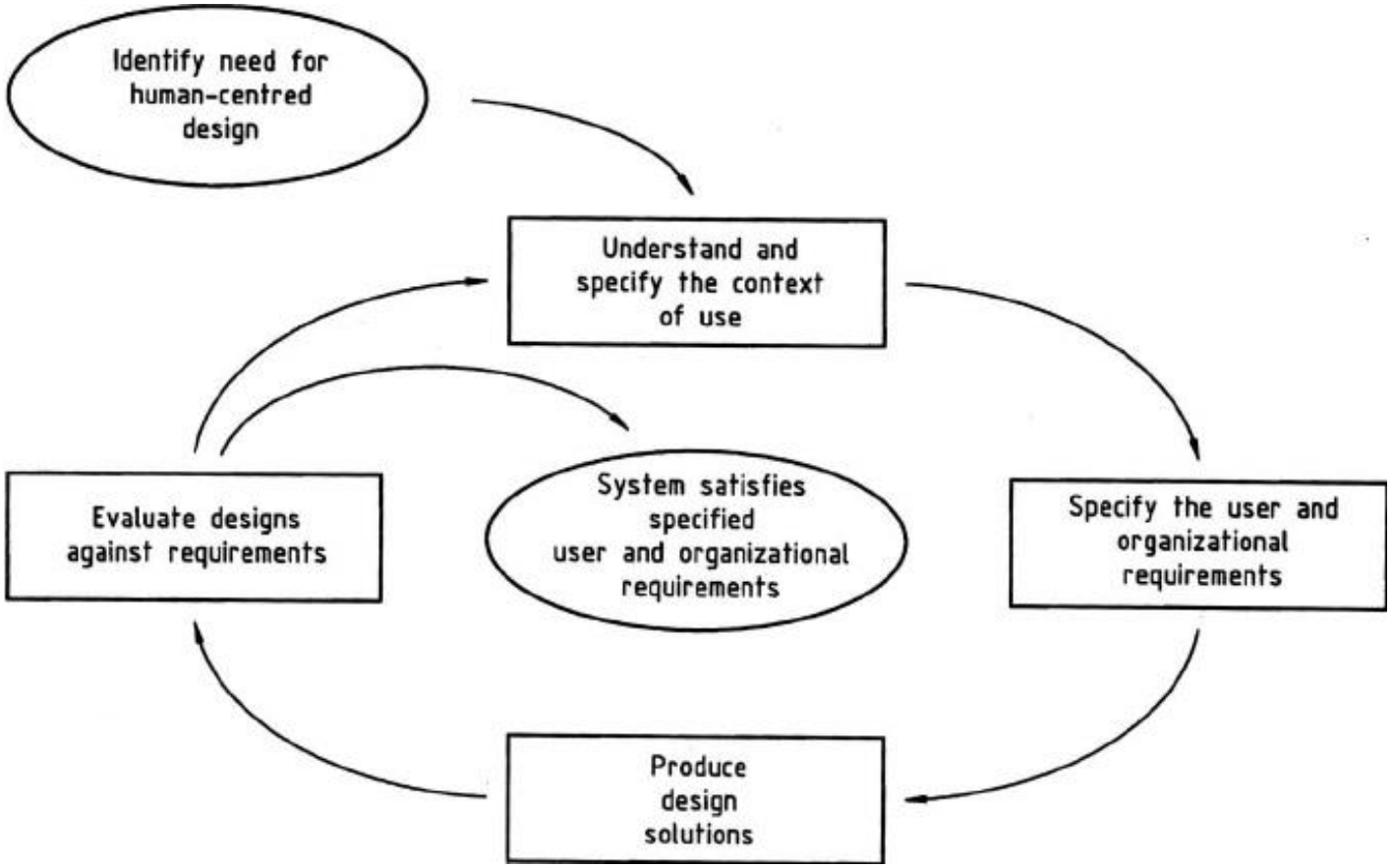
do utilizador

USER-CENTERED DESIGN

7 princípios do Desenho Centrado no utilizador

- Usar o conhecimento "na cabeça" e disseminado no ambiente
- Simplificar a estrutura das tarefas
- Tornar as coisas **visíveis**
- Fazer os **mapeamentos** certos
- Explorar o poder das **restrições**
- Desenhar para o **erro**
- Quando tudo o resto falhar, normalizar

TRABALHO RELACIONADO



Maguire (2001)

TRABALHO RELACIONADO

Etapa 1: Planeamento do processo centrado no utilizador	Etapa 2: Compreender e especificar o contexto de uso	Etapa 3: Especificar o utilizador e organizar os requisitos	Etapa 4: Propor uma solução de design	Etapa 5: Avaliar a solução de design com os requisitos
1. Planeamento de utilização 2. Análise do custo benefício	1.Identificação dos stakeholders 2.Análise do contexto de uso 3.Analise dos utilizadores que usam aplicação **4.Analise dos utilizadores através da observação** 5.Manter o registo da utilização 6.Análise das operações	1.Análise dos stakeholeders 2.Análise do custo benefício para o utilizador **3.Entrevistas com os utilizadores** **4.Focus group** 5.Casos de uso **6.Personas** 7.Sistema existente; Análise de competidores 8.Tarefa/função mapeamento 9.Alocação de funções 10.Utilizador, usabilidade e organização de requisitos	1.Brainstorming 2.Design paralelo 3.Diretrizes de projeto e padrões 4.Storyboard 5.Diagrama de afinidade **6.Sorting Card** **7.Prototipagem** **8.Wizard-of-Oz**	1.Avaliação participativa 2.Avaliação assistida **3.Av. Heuristica** 4.Controlo dos testes dos utilizadores **5.Questionário de satisfação** **6.Avaliação cognitiva carga de trabalho** 7.Incidentes críticos **8.Entrevistas a posteriori**