



# ANÁLISE E PROJETO DE SISTEMAS

---






Engenharia de Software

TeSP - Redes e Segurança Informática

# ENGENHARIA DE SOFTWARE

## ESTRUTURA

---

-  Software
-  Sistema e produto de software
-  Sistemas de informação
-  Engenharia de software
-  Processo de software



# ENGENHARIA DE SOFTWARE

## SOFTWARE

---

### Software

- O termo '*software*' foi usado a primeira vez em 1959;
- A Applied Data Research, fundada no mesmo ano, foi a primeira empresa a vender *software* de forma separada do *hardware*;
- Em 1970, a justiça norte-americana exigiu à IBM que distinguísse contabilisticamente o *hardware* do *software*



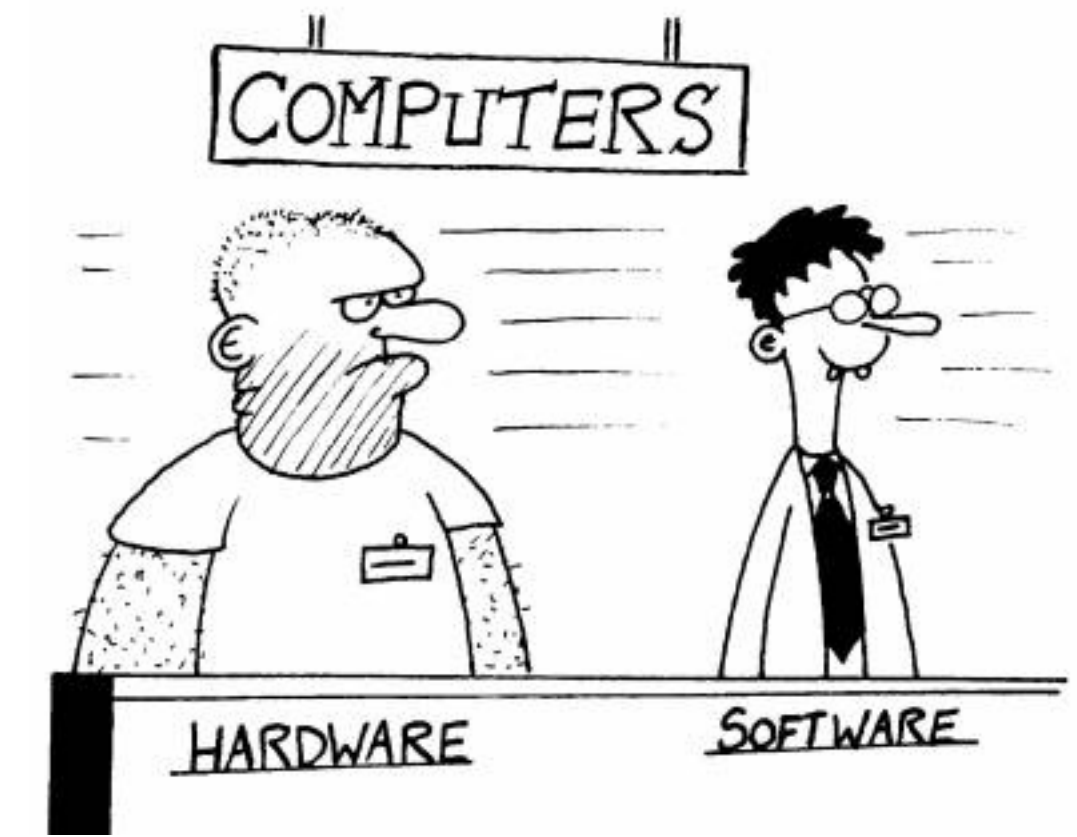
# ENGENHARIA DE SOFTWARE

## SOFTWARE

---

### Software vs. Hardware

- O **hardware** engloba todos os componentes físicos do computador, ou seja, é constituído pelos componentes eletrónicos e eletromecânicos (e.g. processador, memória, discos);
- O **software** é tudo aquilo que compõe um computador, exceptuando o hardware [Galler, 1992], ou seja é a componente lógica do computador que tem a capacidade de comandar o hardware;
- É um conjunto de programas, procedimentos e regras, relativo ao funcionamento de um sistema que visa a aquisição, o armazenamento e o processamento de dados;



# ENGENHARIA DE SOFTWARE

## SOFTWARE

---

### Características do Software

- Natureza intangível
  - Não tem existência física;
  - Só a sua funcionalidade é que é perceptível;
  - Não está condicionado pelas propriedades dos materiais, nem regido pelas leis da física;
  - É maleável, uma vez que é relativamente fácil sujeitá-lo a modificações;
  - É desenvolvido, não é fabricado ou construído;
  - Produzi-lo a primeira vez é caro, no entanto a sua cópia tem muito baixo preço;
  - Copiá-lo é fácil e não diminui a sua qualidade.

# ENGENHARIA DE SOFTWARE

## SOFTWARE

---

### Características do Software

- Ausência de desgaste
  - Não apresenta desgaste, pois não perde qualidades (físicas) ao longo do tempo;
  - A sua degradação provem das alterações que vai sofrendo para mantê-lo útil,
  - A incorporação de novas funcionalidades pode implicar defeitos no software;
  - O software acaba por perder qualidades ao longo da sua vida útil;

# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

### Sistema de software

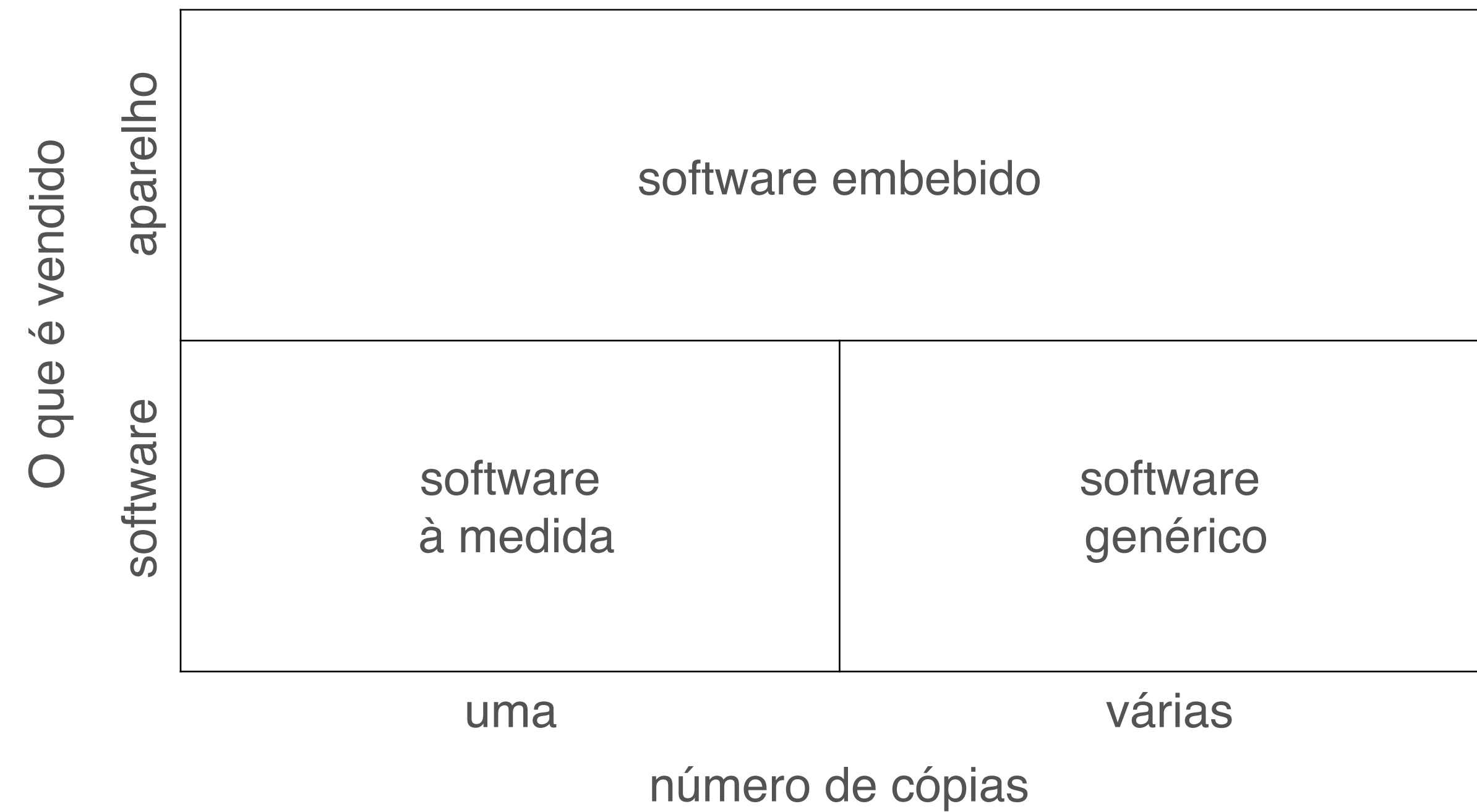
- Um **sistema** é um conjunto identificável e coerente de componentes que interagem entre si para atingir um determinado objetivo;
- Um **sistema de software** é um sistema baseado em software formando uma plataforma de computação - uma combinação de hardware e software;

# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

### Classificação de software





# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

### Classificação do software

- O software incluído em qualquer aparelho/dispositivo é designado de **software embebido**. A não valorização desta tecnologia deve-se ao facto que os utilizadores empregam nas partes mecânicas ou electrónicas desses produtos;
- Quando um sistema usa exclusivamente tecnologias de software, pode ser classificado como:
  - **Software à medida**, é desenvolvido por encomenda, para satisfação das necessidades particulares e específicas de um determinado cliente;
  - **Software genérico**, é produzido para ser comercializado para o público em geral.

# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

Um produto de software é composto por:

- **programas**, cujas instruções quando executadas oferecem as funcionalidades do produto;
- **estruturas de dados**, que possibilitam aos programas ter acesso à informação necessária para a execução;
- **documentação**, que explica como instalar, usar e manter os programas.



# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

Um produto de software pode ser classificado de acordo com a proximidade que tem com o hardware.

- O **software de sistema** é responsável por gerir os recursos de hardware do computador, não sendo usado explicitamente pelo utilizador;
- As **aplicações de software** realizam tarefas úteis para o utilizador. São produtos de software desenvolvidos para suportar a realização de tarefas individuais das pessoas e execução de processos das organizações. Para serem executadas recorrem ao hardware e aos software do sistema;

# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

### Qualidades de um produto de software

*“O produto produzido pelo processo de desenvolvimento tem qualidade se cumprir atempadamente todos os requisitos enunciados pelos intervenientes, se não puser de nenhuma forma em causa a vida humana e se consumir os recursos estritamente necessários para atingir esse fim.”*



# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

### Qualidades de um produto de software

- **Funcionalidade** - o conjunto de funções satisfaz as necessidades explícitas e implícitas para a finalidade a que se destina o produto?
- **Fiabilidade** - O desempenho mantém-se ao longo do tempo e nas condições estabelecidas?
- **Usabilidade** - é fácil de usar?
- **Eficiência** - Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?
- **Manutenibilidade** - Há facilidade para correções, atualizações e alterações?
- **Portabilidade**: É possível utilizar o produto em diversas plataformas?

# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

### Domínio

- A área em que o sistema é explorado designa-se por domínio;
- Um domínio pode ser uma área de negócio, uma coleção de problemas ou uma área do conhecimento com a sua própria terminologia;
- O **domínio do problema** é o contexto onde são sentidas as necessidades que têm que ser satisfeitas com o sistema a desenvolver (e.g. cozinheiro, mesas, talheres, refeição...);
- O **domínio da solução** refere-se às atividades que são executadas e aos artefactos que são manipulados e construídos para se resolver o problema;

# ENGENHARIA DE SOFTWARE

## SISTEMA E PRODUTO SOFTWARE

---

### Exercícios

- O que é que distingue o software de outros produtos?
- Que elementos estão incluídos num produto de software?
- Quais as características que definem um produto de software de qualidade?

# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

---

### Dados

- é o nível mais básico e bruto, que ainda não passou por nenhum processo e nenhuma organização para poderem ser utilizados.

### Informação

- acrescenta contexto e significado aos dados. Dados já processado, que já passaram por algum tipo de organização.

### Conhecimento

- acrescenta significado à informação, através da interpretação de como usar adequadamente a informação.

### Sabedoria

- habilidade de usar o conhecimento adquirido de forma simples e dinâmica.





# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

---

### Sistema de informação (SI)

- é um conjunto organizado de elementos, podendo ser pessoas, dados, actividades ou recursos materiais em geral. Estes elementos interagem entre si para processar informação e divulga-la de forma adequada em função dos objectivos de uma organização.



# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

---

### Alguns dos SI mais referidos

- Sistemas de processamento de transações (OLTP).
- Sistemas de informação de gestão (MIS)
- Sistemas de apoio à decisão (DSS)
- Sistemas de gestão de relacionamento com cliente (CRM)
- etc.

# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

---

### **Os SI suportam a execução das atividades das organizações**

- Ajudam as organizações a cumprirem a sua missão e atingirem os seus objetivos.

### **Essas atividades podem ser de diversas áreas:**

- produção;
- marketing;
- contabilidade, finanças, recursos humanos, etc.

### **A realização dessas atividades envolve diversas entidades:**

- funcionários;
- clientes;
- fornecedores, banca, agências governamentais, etc.

# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

---

### Processo de negócio

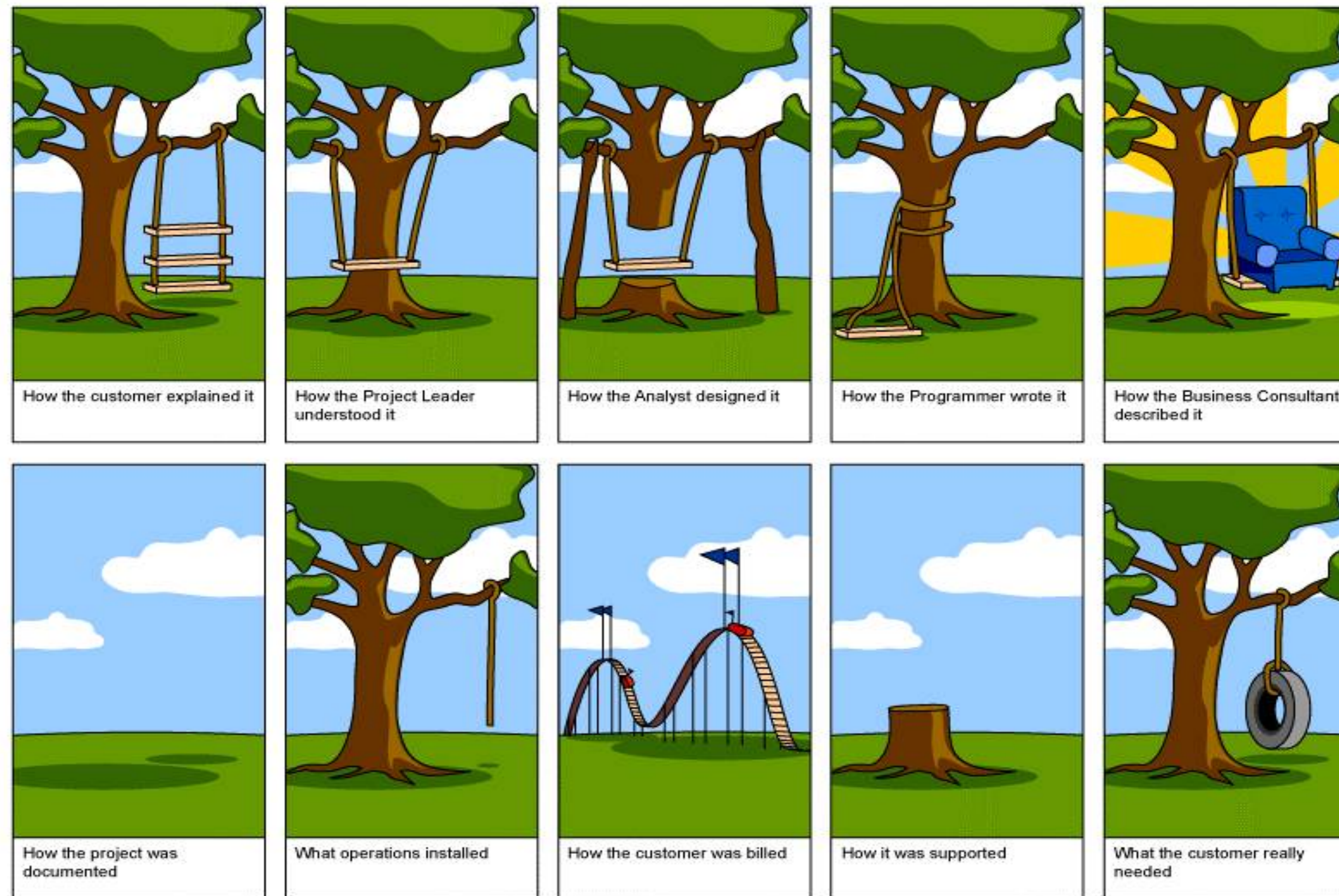
- consiste num conjunto de atividades relacionadas que produzem um serviço ou produto específico;
- são um recurso importante das organizações e devem ser devidamente compreendidos e geridos;
- a sua correta articulação é fundamental para o bom funcionamento da organização, uma vez que pode envolver diferentes: sistemas; aplicações; áreas funcionais; e entidades;
- O papel dos SI é então suportar a boa execução dos processos de negócio da organização



# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

### Desenvolvimento de SI (DSI)



# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

---

### Problemas no desenvolvimento de SI (DSI)

- Cada pessoa tem um visão diferente sobre o que o rodeia. Num projeto de software não é diferente, havendo pontos de vista diferentes entre os vários intervenientes/*stakeholders*;
- logo há necessidade de envolver todos os intervenientes no processo e utilizar metodologias e ferramentas adequadas para ultrapassar as barreiras de comunicação;
- estimativa errada de custos;
- problemas com requisitos;
- produtividade reduzida;
- para ter sucesso no desenvolvimento deve-se fazer um esforço para produzir software de qualidade, cumprir prazos e respeitar orçamentos e custos.



# ENGENHARIA DE SOFTWARE

## SISTEMA DE INFORMAÇÃO

---

### Exercícios

- Para que serve o SI?
- Os processos de negócio são parte do SI (S/N) ? Justifique.
- Indique alguns tipos de SI?
- Quais os maiores problemas no desenvolvimento de SI?



# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

### Engenharia de Software (SE)

- é o ramo da informática que aborda a construção de sistemas de software, cuja complexidade obriga que sejam construídos por uma equipa de engenheiros.

### corresponde à

- aplicação dos princípios da engenharia ao processo de desenvolvimento de software;
- utilização dos princípios básicos da engenharia para obter, de forma economicamente viável, software fiável e que corra eficientemente em computadores reais [Bouer, 1968];

**A engenharia de software não está apenas centrada nos aspectos técnicos, inclui também atividades de gestão do próprio processo de software.**



# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

### Programação de computadores $\neq$ Engenharia de Software

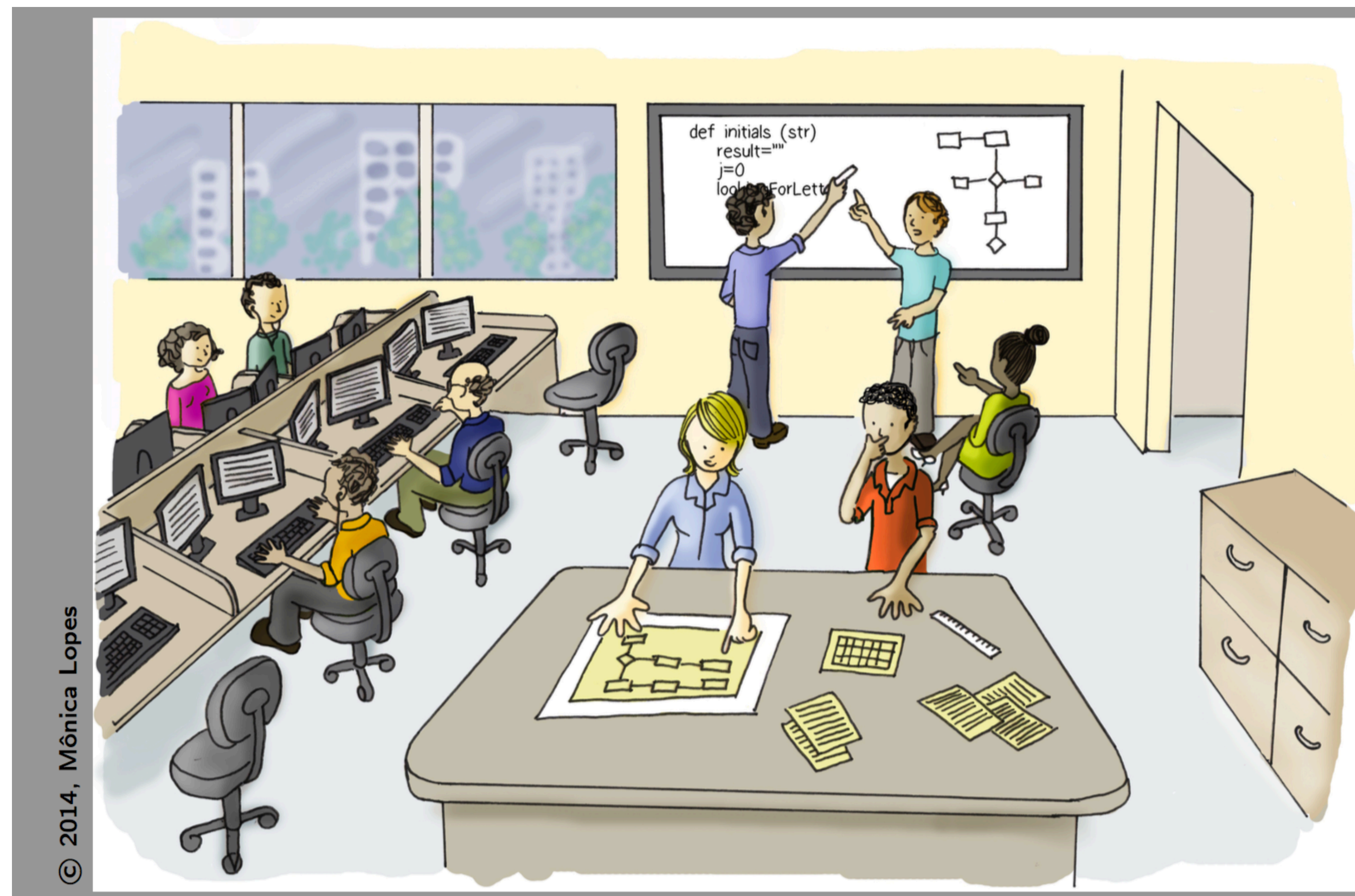
- um programador escreve programas de software;
- é difícil para um programador perceber todas as facetas de um sistema de software;
- um engenheiro de software dedica menos de 10% do seu tempo a programar;
- um engenheiro de software deve ser capaz de gerir o desenvolvimento de sistemas úteis e eficientes;
- os engenheiros de software coordenam tecnicamente o trabalho dos programadores;
- a diferença entre os programadores e os engenheiros de software está relacionada com a responsabilidade profissional;

# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

### Ambiente típico de Engenharia de Software



# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

**Apesar da dimensão e sofisticação da indústria de software, no final de um projeto continuam a surgir questões do género:**

- Porque demorou tanto tempo a concluir?
- O que levou a custos tão elevados?
- Porque é que não se conseguem eliminar todos os erros antes do software entrar em produção?
- Porque é tão caro manter o software existente?
- Porque continua a ser tão difícil medir o progresso no desenvolvimento do software?



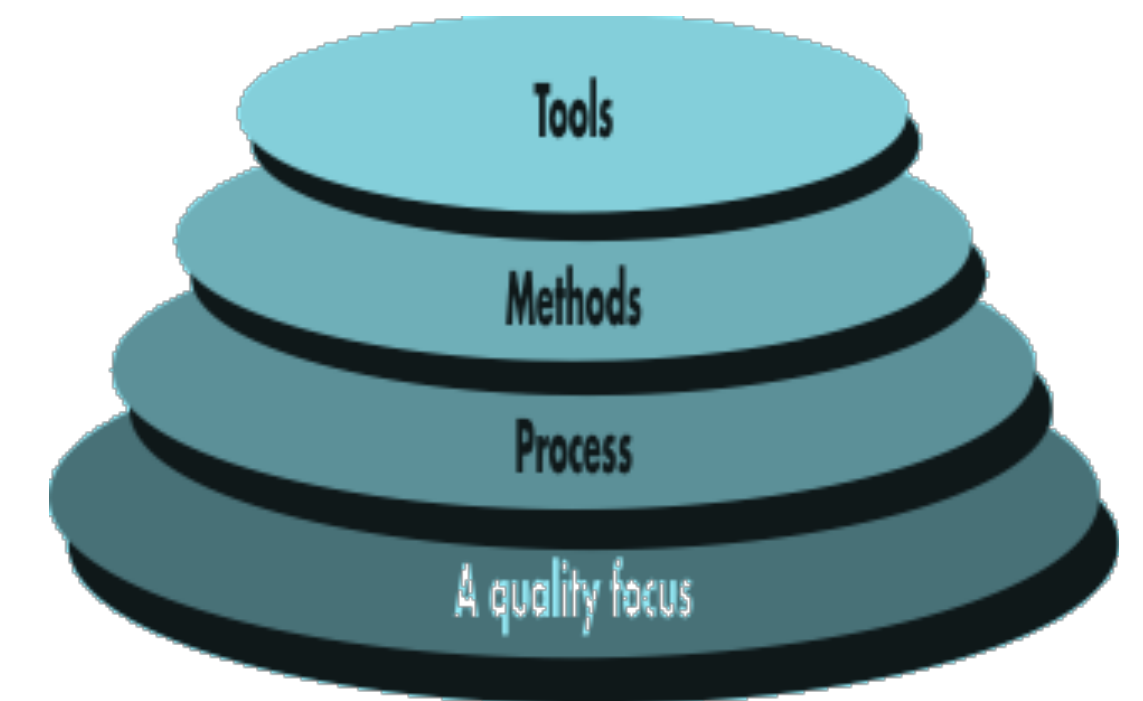
# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

### Camadas da Engenharia de Software

- A engenharia de software tem como principal objetivo a **qualidade**;
- O **processo** define o enquadramento para o realização das tarefas de desenvolvimento e a aplicação dos respetivos métodos;
- Os **métodos** consistem no conhecimento e nas técnicas específicas como a análise de requisitos, modelação ou a codificação;
- As **ferramentas** são os meios auxiliares utilizados. Exemplos: CASE, IDE e SCM.



# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

### Desafios da Engenharia de Software

- **Prazo de entrega**

- Desenvolvimento de técnicas de software que permitam reduzir os prazos de entrega;
- Medir o progresso no desenvolvimento do software e obter estimativas mais precisas.

- **Qualidade**

- Desenvolver técnicas e metodologias para garantia e melhoria da qualidade do software;
- Eliminar os erros antes do software entrar em produção;
- Desenvolver software que seja fácil/barato de manter.

- **Produtividade / Custo**

- Entrega de software de qualidade a custos controlados;
- Perceber a origem dos custos e como otimizar a produção.



# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

### Princípios da Engenharia de Software

- **Acrescentar valor ao cliente**
  - É para isso que tudo existe. Alinhar com o negócio.
- **KISS (Keep It Simple, Stupid!)**
  - A solução deve ser o mais simples possível, mas não a mais simples.
- **Manter os objetivos/visão**
  - Necessidade de forte *sponsor*;
- **Produzir software que outros possam alterar/melhorar**
  - Estruturar e documentar convenientemente.
- **Antecipar a mudança**
  - O software deverá ser flexível para lidar com situações previsíveis.
- **Promover a reutilização do software**
  - O software como um componente que pode ser usado em várias soluções

# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

### Áreas de conhecimento do SWEBOK

- SWEBOK (Software Engineering Body of Knowledge): é um referencial para a caracterização da disciplina de engenharia de software.

1	requisitos de software	9	modelos e métodos de eng. <sup>a</sup> softw.
2	conceção de software	10	qualidade no software
3	construção de software	11	prática profissional em eng. <sup>a</sup> softw.
4	teste de software	12	economia da eng. <sup>a</sup> software
5	manutenção de software	13	fundamentos de informática
6	gestão de configurações de softw.	14	fundamentos de matemática
7	gestão da eng. <sup>a</sup> software	15	fundamentos de engenharia
8	processo de eng. <sup>a</sup> software		

# ENGENHARIA DE SOFTWARE

## ENGENHARIA DE SOFTWARE

---

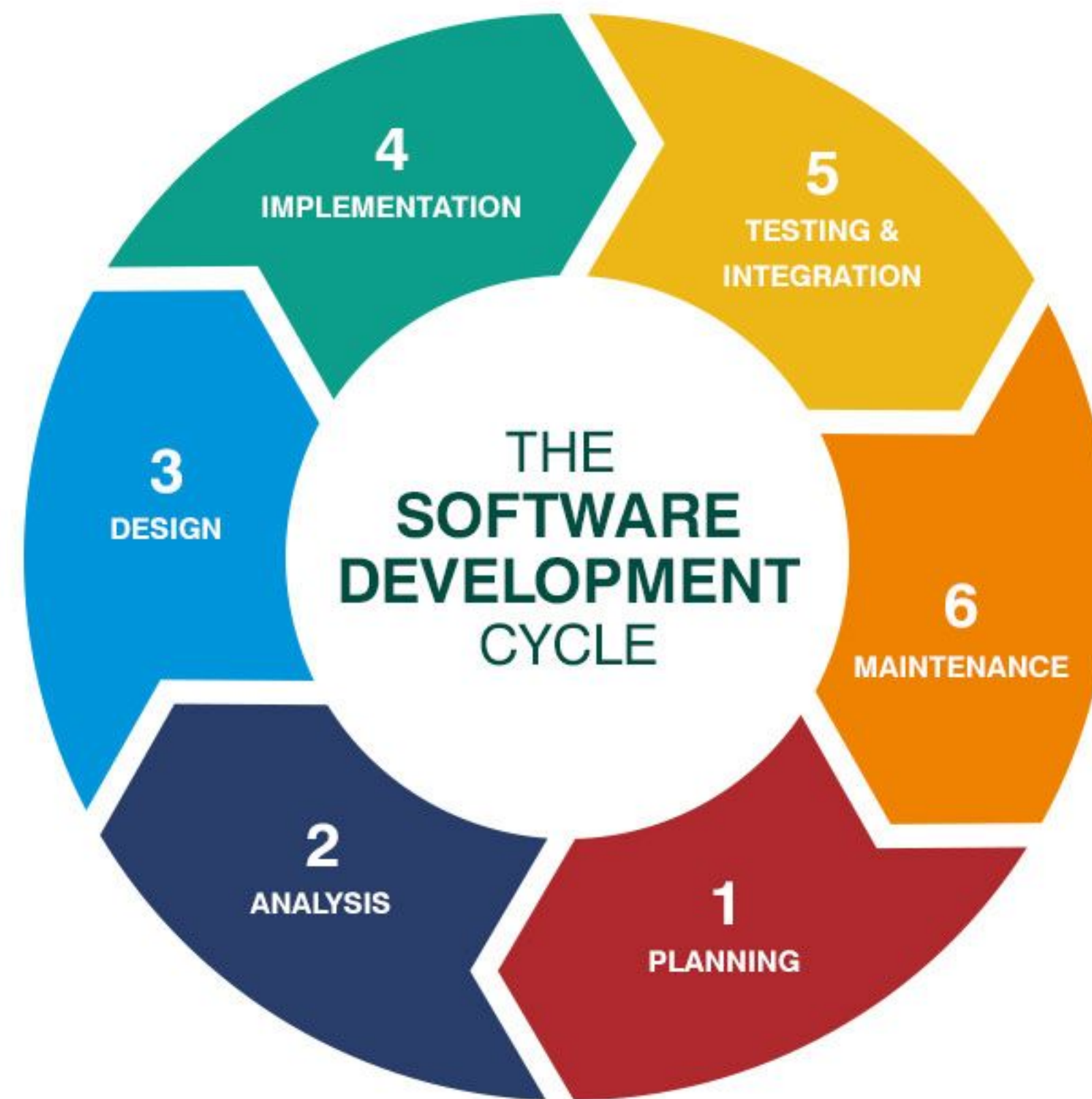
### Exercícios

- Qual o âmbito de aplicação da engenharia de software?
- Quais são os principais desafios da engenharia de software.
- Identifique 3 princípios da engenharia de software?
- Quantas áreas de conhecimento incluem o corpo de conhecimento da engenharia de software de acordo com o SWEBOK.

# APRESENTAÇÃO

## PROCESSO DE SOFTWARE

---

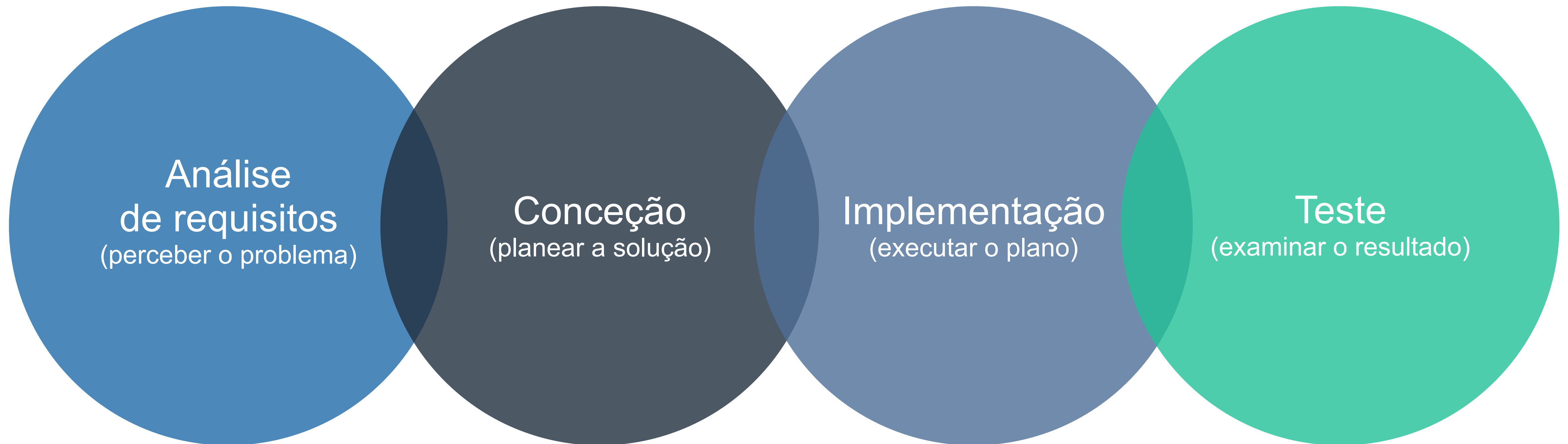


# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

Processo de software





# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

### Modelos do processo de software

- Um modelo de processo representa um processo de desenvolvimento, indicando como este é organizado.
- Os modelos de processo auxiliam os engenheiros no estabelecimento da relação entre as atividades e as técnicas que compõem o processo de desenvolvimento.
- **Ao longo do tempo têm vindo a ser propostos modelos de desenvolvimento de software genéricos por diversos investigadores:**
  - Modelo em Cascata
  - Modelo em V
  - Modelos iterativos
    - Modelo incremental
    - Modelo em espiral
  - Modelos baseados em prototipagem

# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

### Objetivos dos modelos do processo de software

- definir as atividades que devem ser seguidas para desenvolver sistemas;
- introduzir coerência no desenvolvimento, assegurando que os sistemas são desenvolvidos segundo os mesmos princípios;
- fornecer pontos de controlo para avaliar os resultados obtidos e verificar cumprimento de prazos;
- fomentar uma reutilização de componentes, para incrementar a produtividade das equipas no projeto.

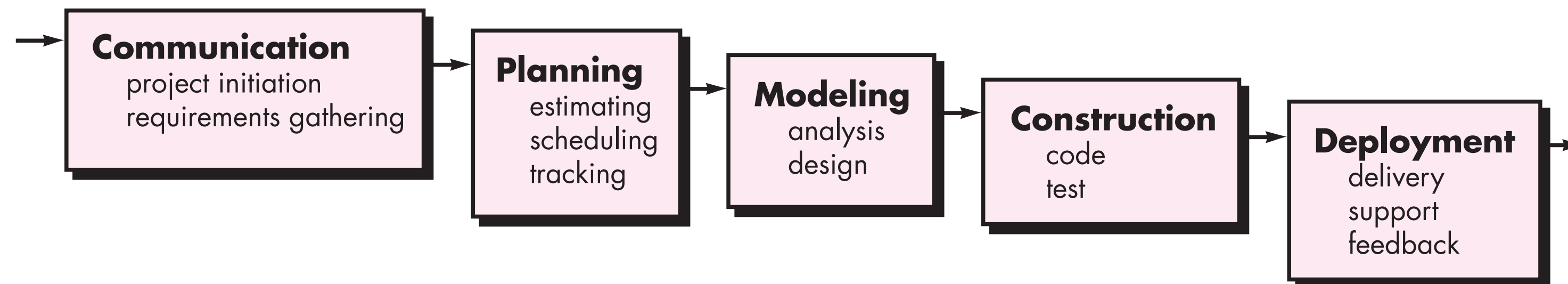
# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

### Modelo em Cascata

- primeiro modelo a ser proposto (1956);
- adequado para grandes projetos;
- facilita a integração de múltiplas equipas;
- exige que requisitos estejam bem definidos e lida mal com a sua mudança;
- só no final do projeto, o sistema pode ser validado.

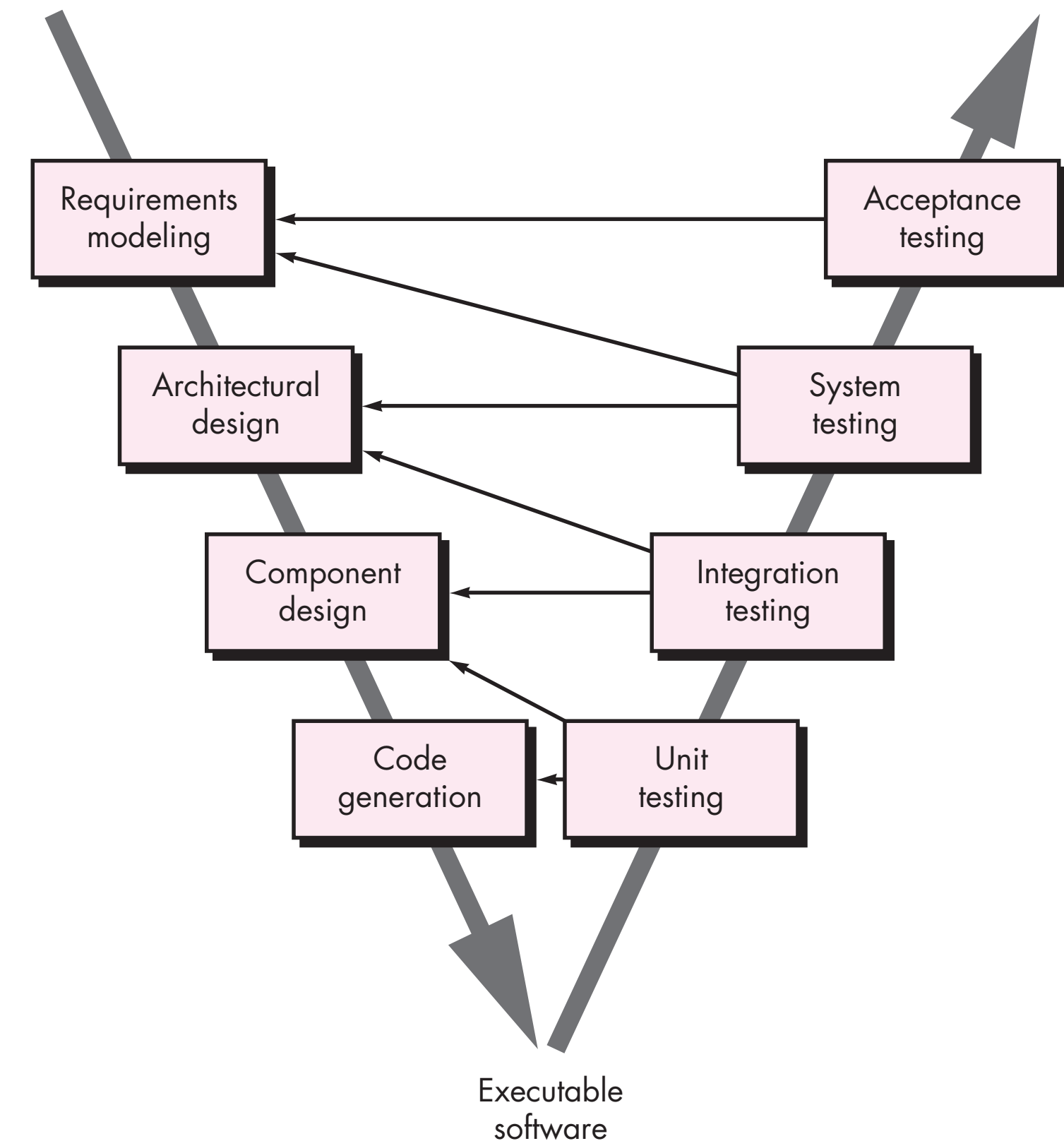


# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

### Modelo de Desenvolvimento em V

- variante do modelo em cascata;
- foca a garantia da qualidade;
- facilita a integração de múltiplas equipas;
- as atividades iniciais elaboram os testes que são realizados na parte final.

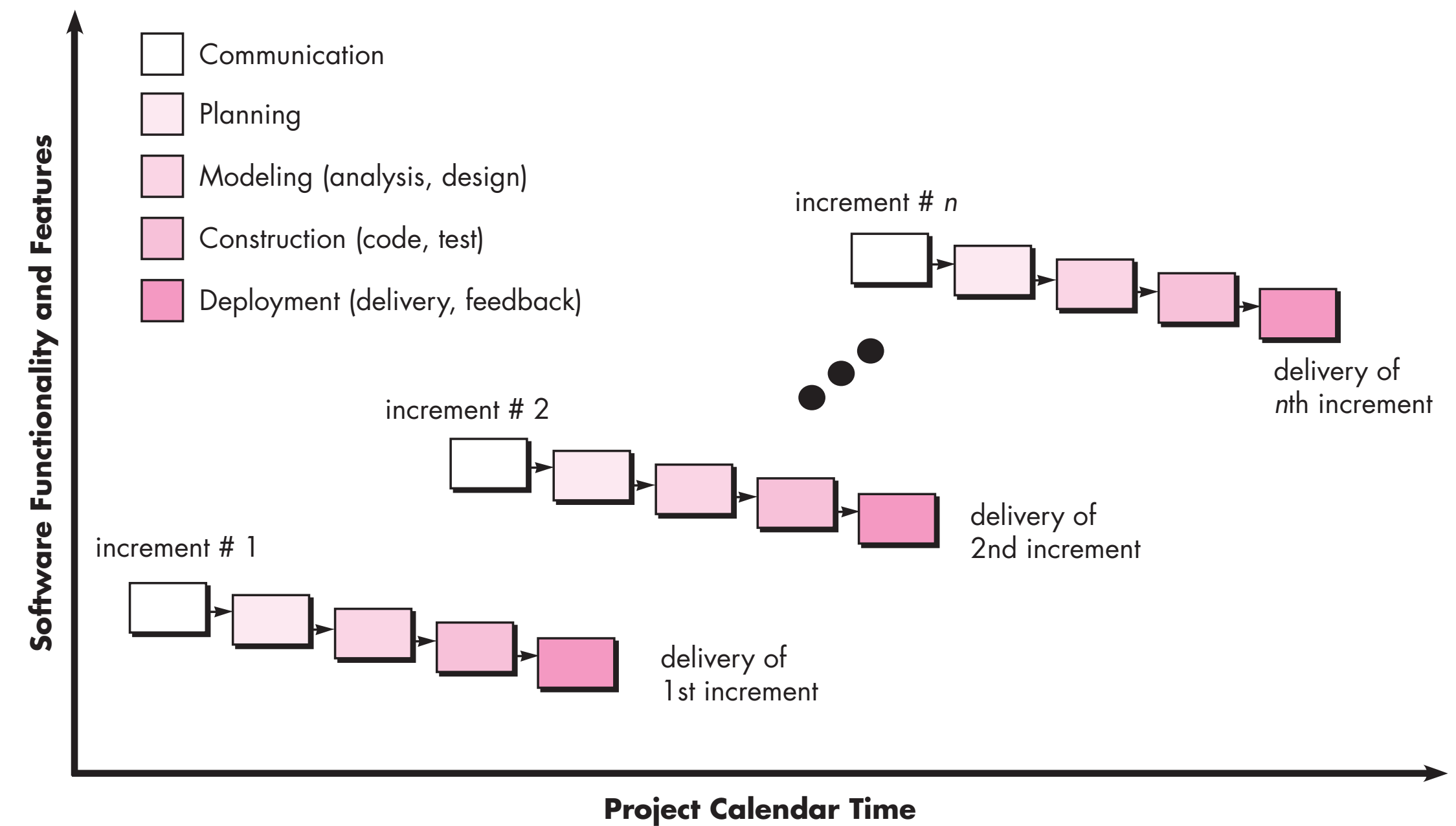


# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

### Modelo Incremental

- tenta resolver alguns problemas do modelo em cascata;
- adia o desenvolvimento das funcionalidades por definir;
- valida antecipadamente cada parte do sistema;
- distribui melhor as tarefas pelas equipas ao longo do tempo.





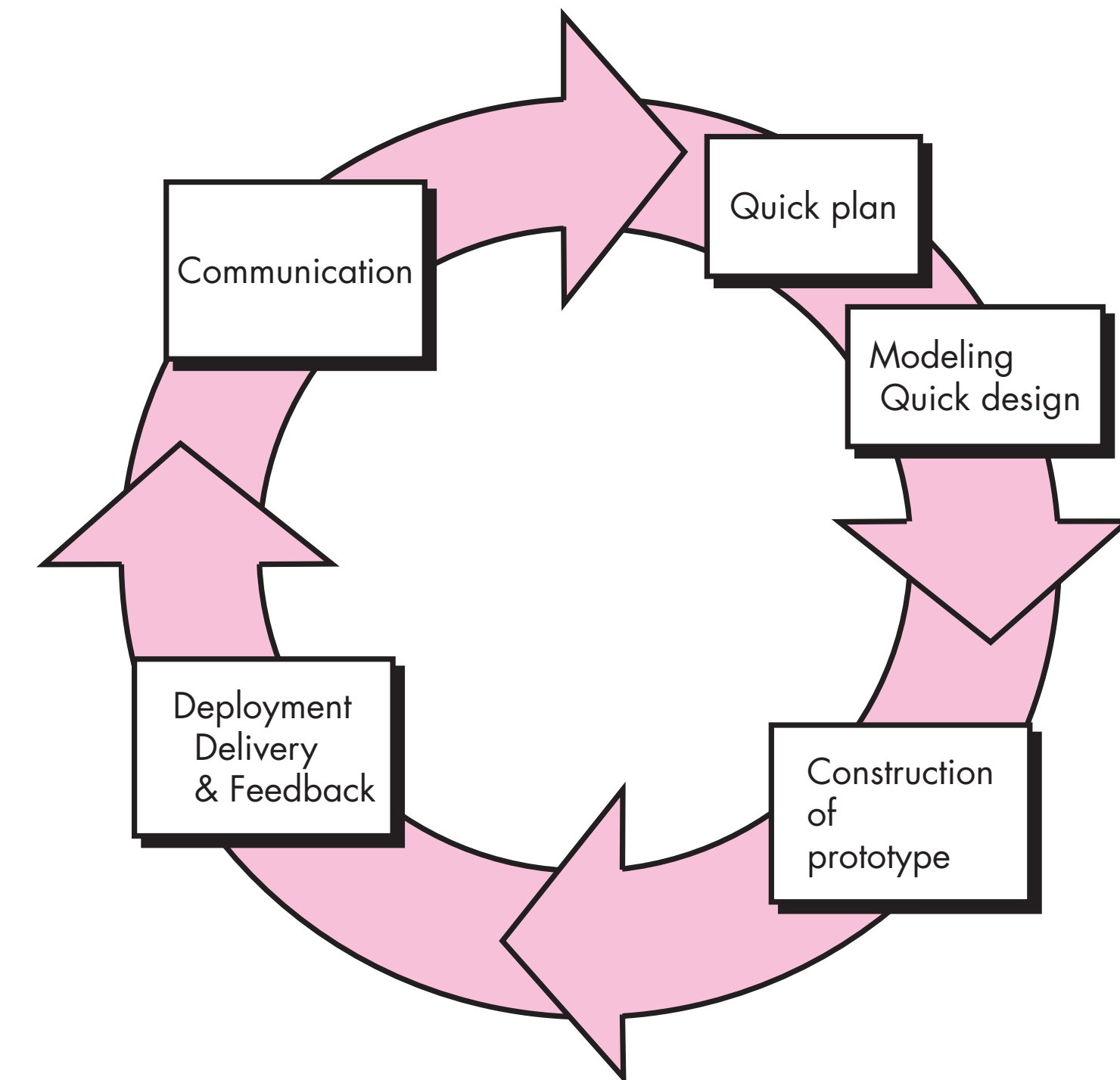
# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

### Modelo baseado na prototipagem

- aplica-se normalmente:
  - quando o cliente não sabe bem o que quer;
  - quando o sistema é inovador e certos conceitos necessitam de ser previamente testados
- pode ser aplicado de forma integrada num dos outros modelos.



# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

### Modelos Ageis

- Foram propostos diversos modelos ágeis para lidar com a mudança de requisitos:
  - eXtreme Programming (XP)
  - Crystal
  - Scrum
- acrescentam uma componente humana ao processo;
- baseiam-se nos seguintes pressupostos:
  - É difícil prever as mudanças nos requisitos durante o SDLC
  - O projeto e a produção do software não são atividades estanques
  - Não é possível planejar detalhadamente a análise, projeto, codificação e teste
- O manifesto ágil, realizado 2001, sintetiza os principais conceitos e valores do desenvolvimento ágil de software.

# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

### Manifesto ágil

- O ***Agile Manifesto*** foi um documento criado em 2001 por um conjunto de 17 signatários que estabelece os valores e princípios fundamentais das metodologias ágeis.
- **estabelece quatro valores orientadores:**
  - Os indivíduos e suas interações são mais importantes que os procedimentos e ferramentas;
  - Software a funcionar é mais importante que acima de documentação completa;
  - A colaboração com o cliente é mais importante que a negociação de contratos;
  - A capacidade de responder a alterações é mais importante que seguir um plano;

# ENGENHARIA DE SOFTWARE

## PROCESSO DE SOFTWARE

---

### Exercícios

- Durante o desenvolvimento de software, dois defeitos similares foram detetados: um durante a fase de requisitos e outro durante a implementação? Qual destas frases tem maior probabilidade de ser verdadeira?
  1. O defeito mais caro de corrigir é o que foi detido na fase de requisitos.
  2. O defeito mais caros de corrigir é o que foi detetado durante a implementação.
  3. O custo de correção dos defeitos é tendencialmente similar.
  4. Não há relação entre a fase em que o defeito é detetado e o custo de reparação.
- Aponte algumas vantagens que resultam da utilização de um processo iterativo e incremental face a um processo sequencial, que segue o modelo em cascata.
- Identifique os valores do manifesto ágil.





# ANÁLISE E PROJETO DE SISTEMAS

---

TeSP - Redes e Segurança Informática