

Engenharia de Software

Introdução à engenharia de software

Organização da apresentação

- História;
- Engenharia de software;
- SWEBOK - áreas de conhecimento da engenharia de software;
- Software;
- Produto de software;
- Ciclo de vida do desenvolvimento de software;

História

- O termo “Engenharia de Software” foi sugerida numa conferência organizada pela NATO em 1968;
- Tinha como objetivo discutir a atual “crise no software”;
- Esta crise prendia-se com as dificuldades no desenvolvimento de sistemas complexos na década de 60;
- Foi proposto a adoção de uma abordagem de engenharia para o desenvolvimento de software de forma a reduzir os custos no desenvolvimento de software de forma a garantir mais fiabilidade no software;
- Daqui resultou um relatório que definia os fundamentos da engenharia de software.



História

at universities. The discussions cover all aspects of software including

- relation of software to the hardware of computers
- design of software
- production, or implementation of software
- distribution of software
- service on software.

By including many direct quotations and exchanges of opinion, the report reflects the lively controversies of the original discussion.

Although much of the discussions were of a detailed technical nature, the report also contains sections reporting on discussions which will be of interest to a much wider audience. This holds for subjects like

- the problems of achieving sufficient reliability in the data systems which are becoming increasingly integrated into the central activities of modern society
- the difficulties of meeting schedules and specifications on large software projects
- the education of software (or data systems) engineers
- the highly controversial question of whether software should be priced separately from hardware.

Caracterização da disciplina

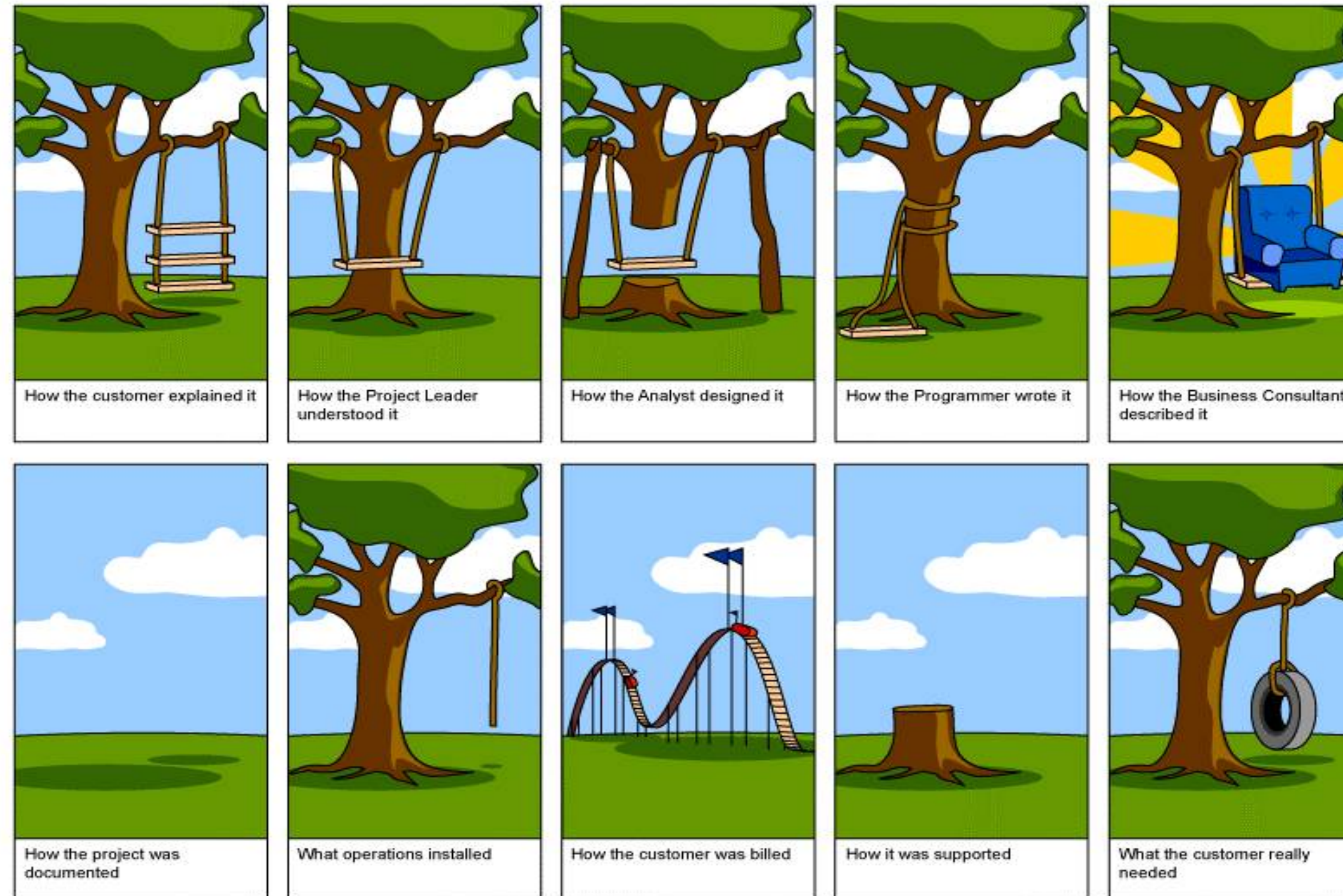
- A engenharia de software corresponde à:
 - ✓ aplicação dos princípios da engenharia tradicional ao processo de desenvolvimento de software;
 - ✓ utiliza os princípios básicos da engenharia para obter, de forma economicamente viável, software fiável e eficiente - que corra eficientemente em computadores reais [Bauer, 1968].
 - ✓ aplicação duma abordagem sistemática, disciplinada e quantificável no contexto do planeamento, do desenvolvimento e da exploração de sistemas de software [PRESSMAN, 2016].

Caracterização da disciplina

- A engenharia de software não está centrada apenas nos aspetos técnicos associados ao desenvolvimento, mas também inclui atividades de gestão do próprio processo de desenvolvimento.
- Num projeto de com uma equipa numerosa, existe a necessidade de gerir, planear e controlar as atividades do vários profissionais.



Desenvolvimento de software



Engenharia de software - Desafios

- Prazos 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 - Princípios

- 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;
- 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.

Exercícios

1. Qual o âmbito de aplicação da engenharia de software?
2. Assinale a alternativa que preenche as lacunas corretamente relativa a definição abaixo para Engenharia de Software.

De acordo com Pressman a Engenharia de Software é a aplicação de uma abordagem sistemática, _____ e quantificável no contexto do planeamento, do _____ e exploração de sistemas de software.

- A. disciplinada – desenvolvimento.
 - B. completa – implementação.
 - C. incremental – documentação.
 - D. estruturada – implementação.
3. Quais são os principais desafios da engenharia de software?
 4. Identifique e explique 3 princípios da engenharia de software.

Engenharia de software - Camadas (Pressman)

- Segundo Pressman, a engenharia de software é baseada em camada, com foco na qualidade:
 - ✓ **Qualidade:** o principal objetivo da engenharia de software é a qualidade ;
 - ✓ **Processo:** define o enquadramento para a realização das tarefas de desenvolvimento e a aplicação dos respetivos métodos;
 - ✓ **Métodos:** consiste no conhecimento e nas técnicas específicas como a análise de requisitos, modelação e codificação;
 - ✓ **Ferramentas:** são os meios auxiliares usados, e.g. IDE, CASE, SCM.



Engenharia de software - Camadas (Pressman)

- **Processo:**

- ✓ **quem** vai fazer **o quê** e **quando?**

- ✓ constitui a base para o controlo de toda a gestão no processo de desenvolvimento de software;

- ✓ define a metodologia (framework, paradigma);

- ✓ estabelece o contexto no qual são aplicados métodos;

- ✓ são produzidos artefactos , tais como modelos, documentos, relatórios, etc.;

- ✓ são estabelecidos marcos;

- ✓ a qualidade é garantida e mudanças são geridas de forma apropriada (gestão de risco).

Engenharia de software - Camadas (Pressman)

- **Métodos:**

- ✓ fornecem informação técnica para o desenvolvimento o software;
- ✓ definem o conjunto de tarefas com determinadas técnicas para cada fase do desenvolvimento do software;
- ✓ envolvem uma grande variedade de tarefas, que incluem:
 - ✓ comunicação;
 - ✓ análise de requisitos;
 - ✓ design do projeto;
 - ✓ implementação;
 - ✓ testes;
 - ✓ suporte.

Engenharia de software - Camadas (Pressman)

- **Ferramentas:**

- ✓ fornecem suporte automatizado ou semiautomatizado para as camadas de processo e para os métodos;
- ✓ quando as ferramentas são integradas, de modo a garantir que as informações recolhidas por uma ferramenta possam ser utilizadas por outra, é estabelecido um sistema para o suporte ao desenvolvimento de software, denominado CASE
 - Computer-Aided Software Engineering.

Exercícios

1. De acordo com Pressman, a engenharia de software é baseada em camadas, com foco na qualidade. Essas camadas são:
 - a) métodos, processo e testes.
 - b) ferramentas, métodos e processos.
 - c) métodos, construção, teste e implementação.
 - d) planeamento, modelação, construção, validação e implementação.
 - e) comunicação, planeamento, modelação, construção e implementação.

SWEBOK

- O SWEBOK - Software Engineering Body of Knowledge, é um referencial para a caracterização da disciplina de engenharia de software;
- Foi criado pelo IEEE Computer Society;

Áreas de conhecimento

1	requisitos de software	9	modelos e métodos de engenharia de software
2	conceção de software	10	qualidade no software
3	construção de software	11	prática profissional em engenharia de software
4	testes de software	12	economia da engenharia de software
5	manutenção de software	13	fundamentos de informática
6	gestão de configuração de software	14	fundamentos de matemática
7	gestão da engenharia de software	15	fundamentos de engenharia
8	processo da engenharia de software		

SWEBOK I AC1: requisitos de software

- Trata o levantamento, a neles, documentação, validação e a manutenção de requisitos de software;
- Os requisitos exprimem as necessidades e as restrições que são colocados a um sistema de software (ainda em projeto);
- O resultado deste processo é:
 - ✓ o documento de requisitos.

SWEBOK I AC2: conceção de software

- Define a arquitetura, os componentes, as interfaces e outras características do sistema;
- Apoia-se nos requisitos para produzir uma descrição da estrutura interna e da organização do sistema;
- O resultado deste processo deve descrever:
 - ✓ a arquitetura do sistema;
 - ✓ as interfaces entre componentes;
 - ✓ os próprios componentes com pormenor suficiente para permitir a sua construção.

SWEBOK I AC3: construção de software

- Consiste na implementação do software de acordo com o pretendido e a funcionar corretamente;
- Transforma em código as arquiteturas concebidas e descritas na concepção;
- O resultado deste processo é:
 - ✓ o código fonte.

SWEBOK I AC4: teste de software

- É uma parte obrigatória no desenvolvimento de software;
- Avalia-se à qualidade do sistema e procede-se à sua melhoria, identificando defeitos e potenciais problemas;
- Inclui a verificação do comportamento dos sistemas em relação ao funcionamento esperado;
- A verificação decorre a um conjunto casos teste (documento de testes) escolhidos para cobrir as situações mais críticas.

SWEBOK I AC5: manutenção de software

- Introduz modificações no sistema de software, já em produção, com objetivo de:
 - ✓ melhorar o sistema;
 - ✓ corrigir defeitos;
 - ✓ adaptar o sistema a um novo contexto.
- Lida com os defeitos, alterações tecnológicas e as evoluções dos requisitos de utilizador.

“A manutenção consome normalmente 40 a 80% dos custos de software.

É provavelmente a mais importante fase do ciclo do desenvolvimento de software”

Rober L. Glass

SWEBOK I AC6: gestão de configurações de software

- Ou gestão de versões;
- Trata do controlo das alterações e mantém a integridade e a rastreabilidade do sistema de software;
- Inclui atividades de controlo e monitorização que iniciam quando o projeto arranca e que terminam somente quando o sistema é retirado de utilização.

SWEBOK I AC7: gestão da engenharia de software

- Inclui atividades de gestão de software, tais como: planeamento; coordenação; medição, monitorização, controlo e comunicação, para que o sistema seja desenvolvido de forma sistemática, disciplinada e quantificável;
- A gestão praticada em processos de engenharia de outras atividades é diferente da gestão na engenharia de software devido às particularidades do software e dos respetivos processos.

SWEBOK I AC8: processo de engenharia de software

- Proporciona um conjunto de diretivas que instruem a forma como os profissionais devem organizar e executar as suas atividades ao longo do tempo, a nível do desenvolvimento e manutenção de software;
- Permite aferir e melhorar o próprio processo de engenharia de software.

SWEBOK I AC9: modelos e métodos de engenharia de software

- O recurso a modelos e métodos de engenharia de software é fundamental para criar de forma sistemática, disciplinada, quantificável e eficiente sistemas de software;
- Os modelos constituem uma ferramenta indispensável na tomada de decisão em todas as fases do desenvolvimento;

SWEBOK I AC10: qualidade no software

- A preocupação com a qualidade do software deve ser constante;
- A qualidade está ligada à conformidade do sistema relativamente aos requisitos;
- Estende-se por todo o processo de desenvolvimento de software;
- Deve ser visto como uma preocupação transversal a todo o processo de software.

SWEBOK I AC11-15

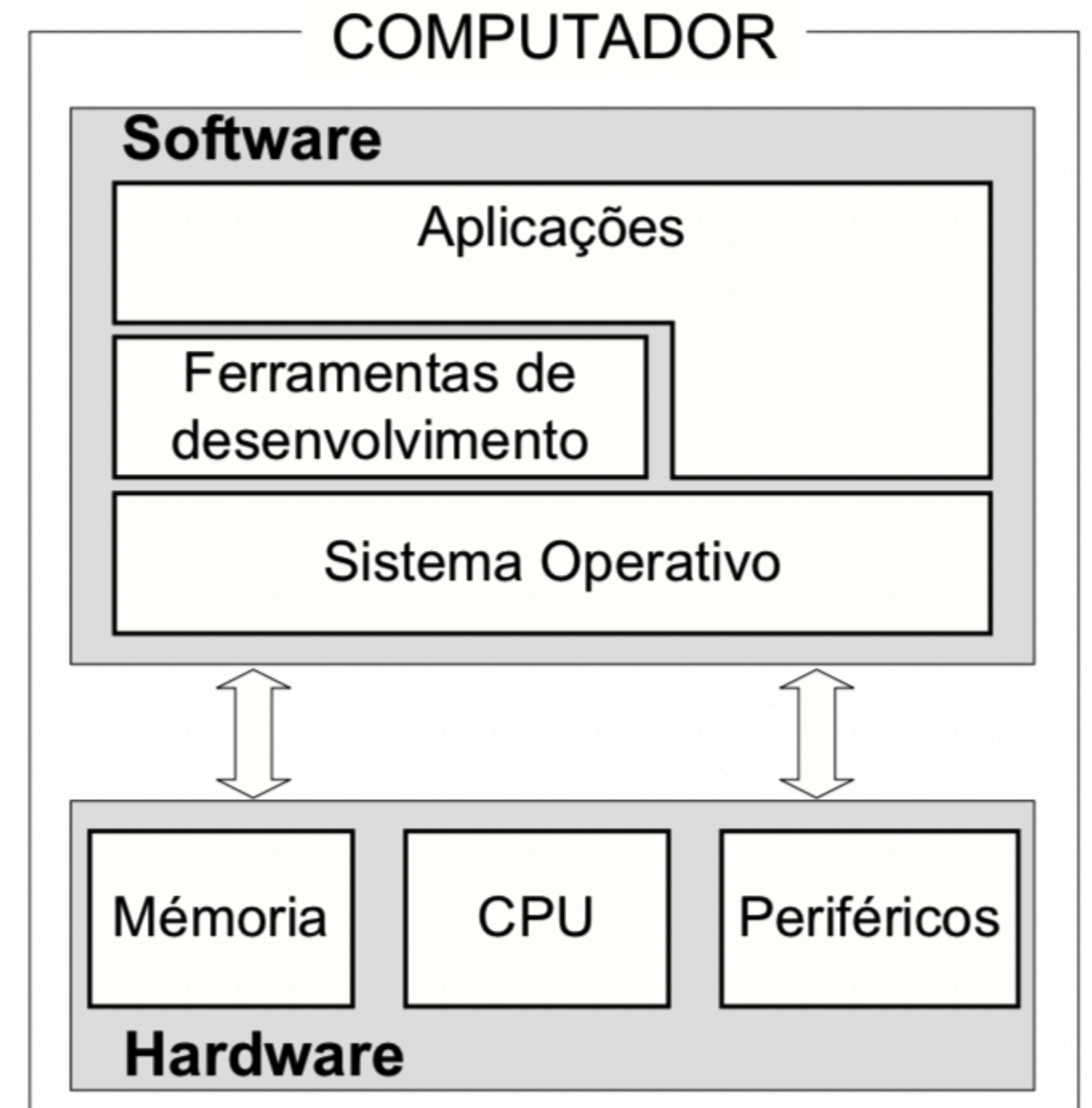
- AC11 - Prática profissional em engenharia de software prende-se com questões relacionadas com o profissionalismo, tais como: ética; dinâmicas de grupo; comunicação; entre outras.
- AC12 - Economia da engenharia de software agrega conteúdos de natureza económica, no âmbito do ciclo de vida dos sistemas de software.
- AC3-15 estão relacionadas com conceitos fundamentais de três disciplinas: informática; matemática e engenharia.

Exercícios

1. Quantas áreas de conhecimento incluem o corpo de conhecimento da engenharia de software de acordo com o SWEBOK
2. Explique em que consistem duas das áreas de conhecimento do SWBOK e explique a sua importância.

Software

- O termo “software” foi usado a primeira vez em 1959;
- A Applied Data Reseach foi a primeira empresa a vender software separadamente do hardware - 1959;
- Tudo que na perspetiva do utilizador, compõe um computador, exceto o hardware, é software [Galler, 1962]



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.

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 - fica obsoleto.

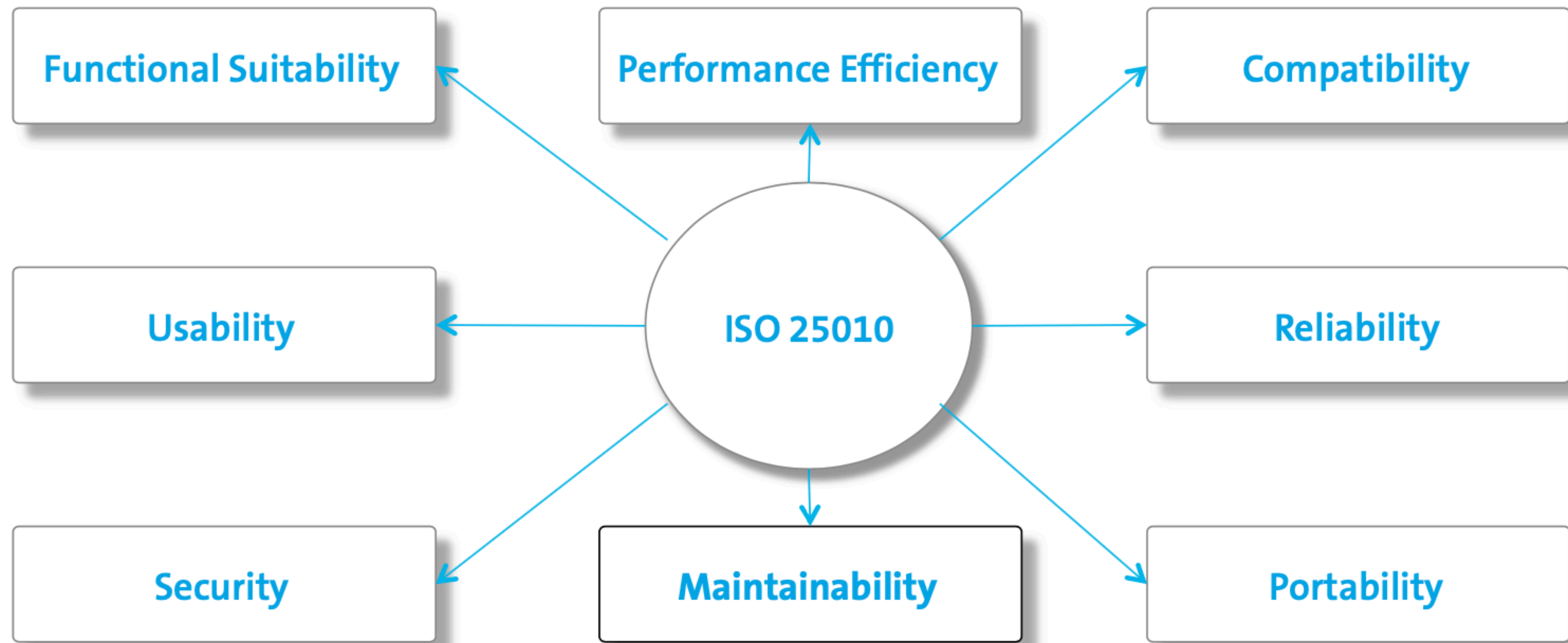
Produto de software

- Um produto de software é composto por:
 - ✓ **programas**: cujas instruções quando executadas oferecem as funcionalidades ao produto;
 - ✓ **estruturas de dados**: que possibilitam aos programas ter acesso a informação necessária para a sua execução;
 - ✓ **documentação**: que explica como instalar, usar e manter os programas.
- Um produto de software pode ser classificado quanto à proximidade que tem com hardware:
 - ✓ **Software de sistema**: responsável por gerir os recursos de hardware do computador, não sendo explicitamente usado pelo utilizador;
 - ✓ **Aplicações de software**: realizam tarefas úteis para o utilizador, sendo desenvolvidos para automatizar tarefas individuais ou processos de organizações. Para serem executadas dependem do software do sistema.

Produto de software - Qualidade (ISO/IEC 9126)

- **Funcionalidade** - o conjunto de funções satisfaz as necessidades explícitas e implícitas para a finalidade a que se destina o produto?
- **Confiabilidade** - Funciona de forma correta, com os recursos pré estabelecidos, é seguro?
- **Usabilidade** - é fácil de usar?
- **Eficiência** - Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?
- **Manutenção** - Há facilidade para correções, atualizações e alterações?
- **Portabilidade**: É possível utilizar o produto em diversas plataformas?

Produto de software - Qualidades (ISO/IEC 25010)



Produto de software - Qualidades (McCall)



Produto de software - Qualidade (McCall)

- Os factores de qualidade segundo McCall concentram-se em três aspetos relativamente a um produto de software:
 - ✓ Características operacionais;
 - ✓ Capacidade de suportar mudanças
 - ✓ Adaptação a novos ambientes

Produto de software - Qualidade (McCall)

- Características operacionais
 - ✓ correção: capacidade do software satisfazer a especificação e os objetivos requeridos pelo cliente;
 - ✓ confiabilidade: capacidade do software de cobrir a funcionalidade pretendida com a precisão exigida;
 - ✓ eficiência: determina a quantidade de recursos exigidos pelo software para desempenhar a sua função;
 - ✓ integridade: controlo de acessos a dados;
 - ✓ usabilidade: esforço necessário para aprender e começar a usar o software;
 - ✓ facilidade de manutenção: esforço necessário para localizar e corrigir erros do software;

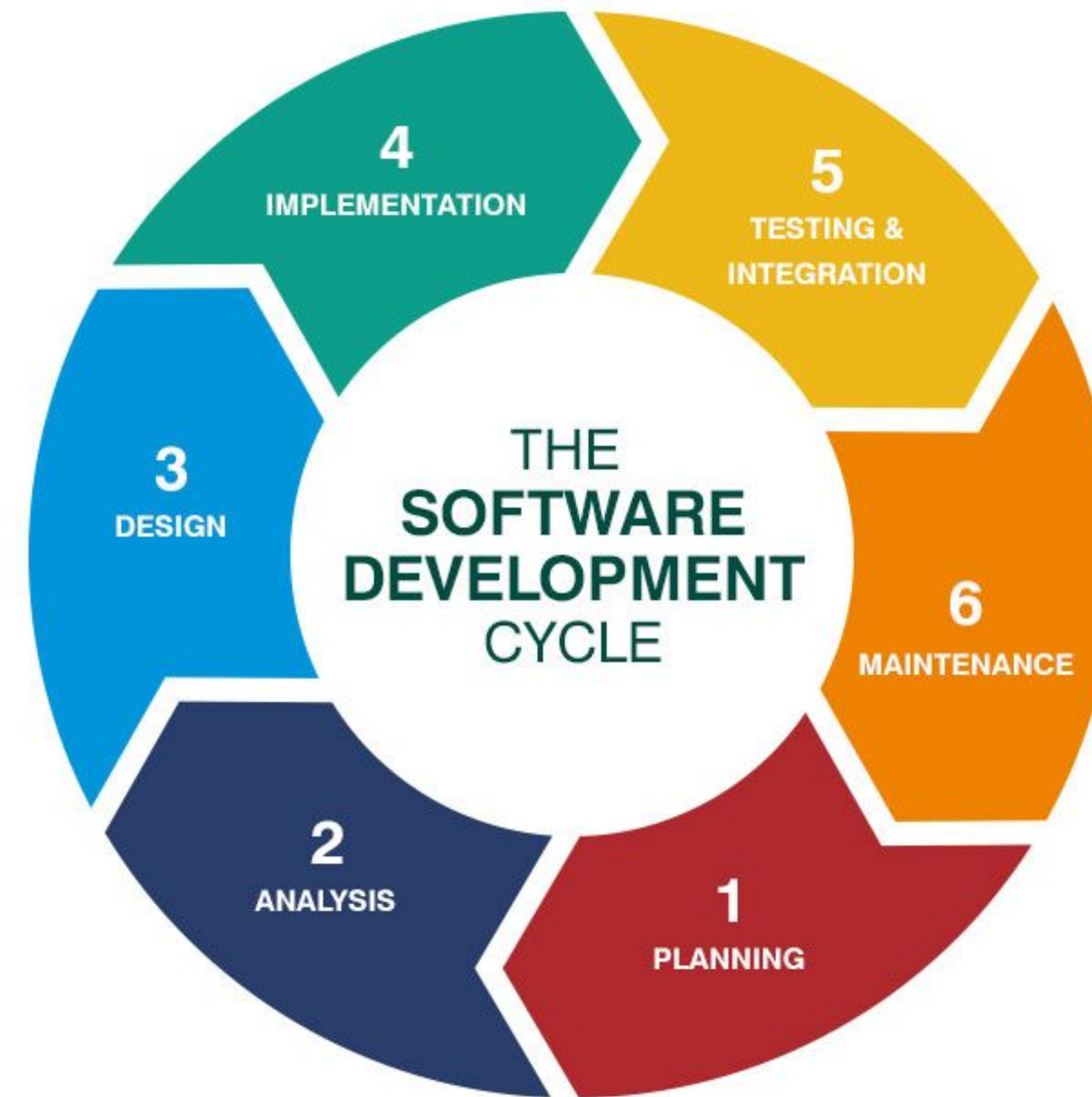
Produto de software - Qualidade (McCall)

- Capacidade de aceitar mudanças
 - ✓ facilidade de manutenção: esforço necessário para localizar e corrigir erros do software;
 - ✓ flexibilidade: esforço necessário para modificar um programa já em produção;
 - ✓ testar: esforço necessário para testar um programa

Produto de software - Qualidade (McCall)

- Adaptação a novos ambientes
 - ✓ portabilidade: capacidade de transferir o software de um ambiente de hardware/software para outro;
 - ✓ interoperabilidade: capacidade de integrar outros sistemas.
 - ✓ reusabilidade: capacidade de reutilizar módulos do software noutras aplicações.

Ciclo de vida do desenvolvimento do software



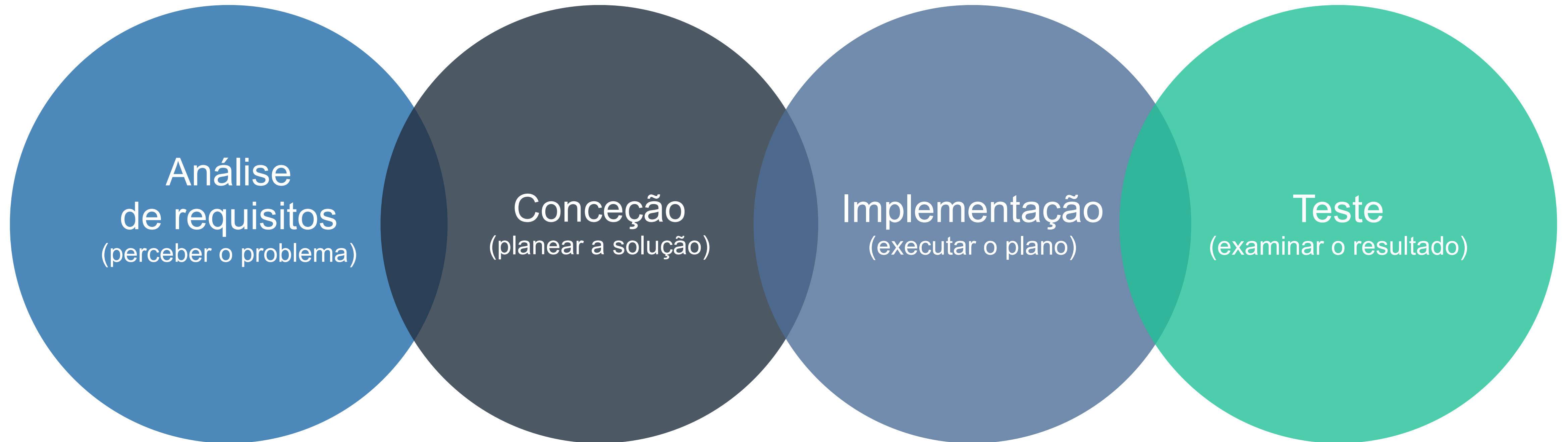
Ciclo de vida do desenvolvimento do software

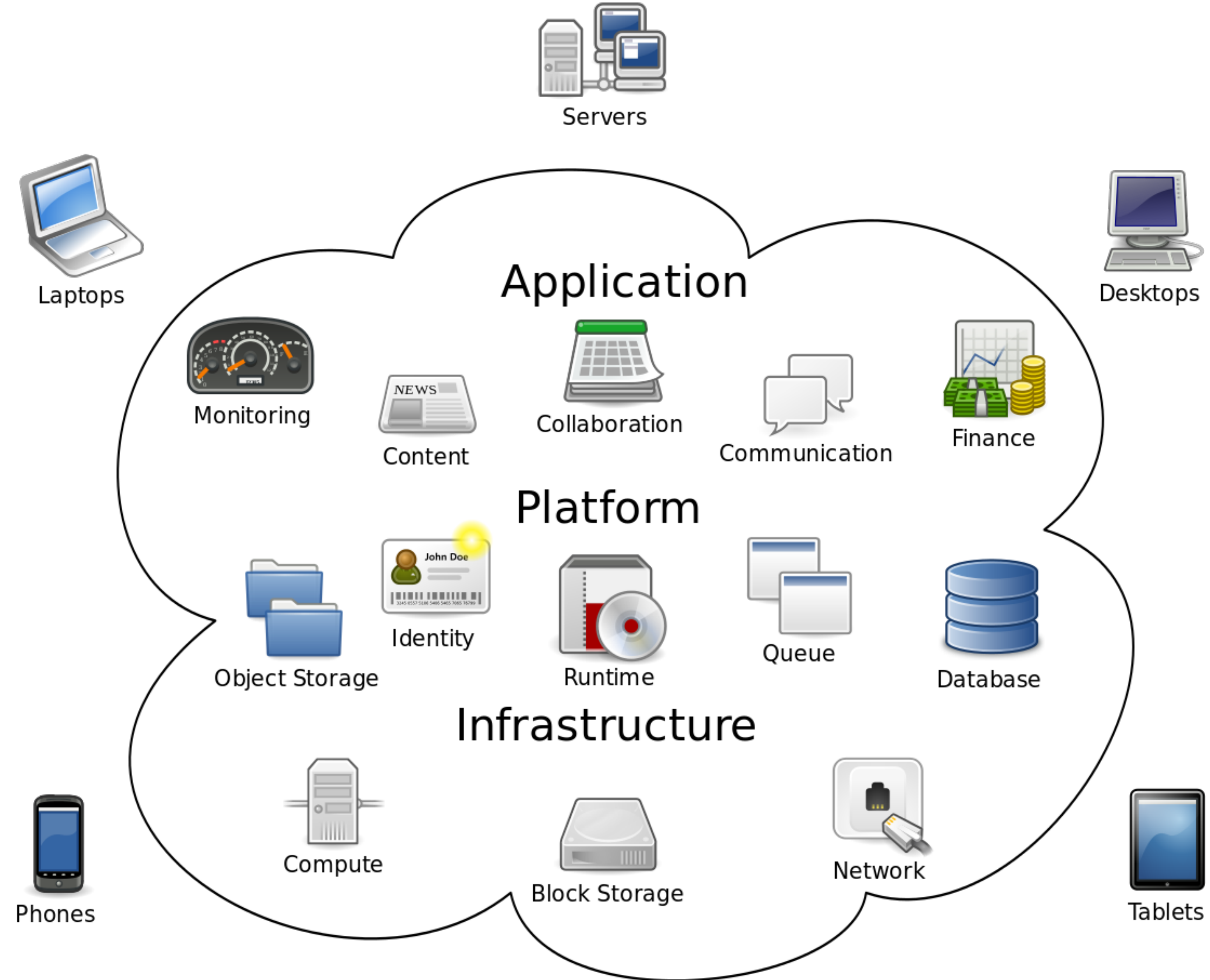
- **Planeamento:** consiste na identificação de problemas, oportunidades e objetivos. Determina ainda os recursos humanos necessários para o projeto.
- **Análise:** consiste no processo de levantamento de requisitos.
- **Desenho:** consiste no desenho e modelação do software que vai ser implementado.
- **Implementação:** consiste no desenvolvimento e documentação de código para desenvolvimento do software.
- **Testes e integração:** o código desenvolvido deve ser testado antes de entrar em produção.
- **Manutenção:** consiste na manutenção e evolução do software até ao fim do seu ciclo de vida.

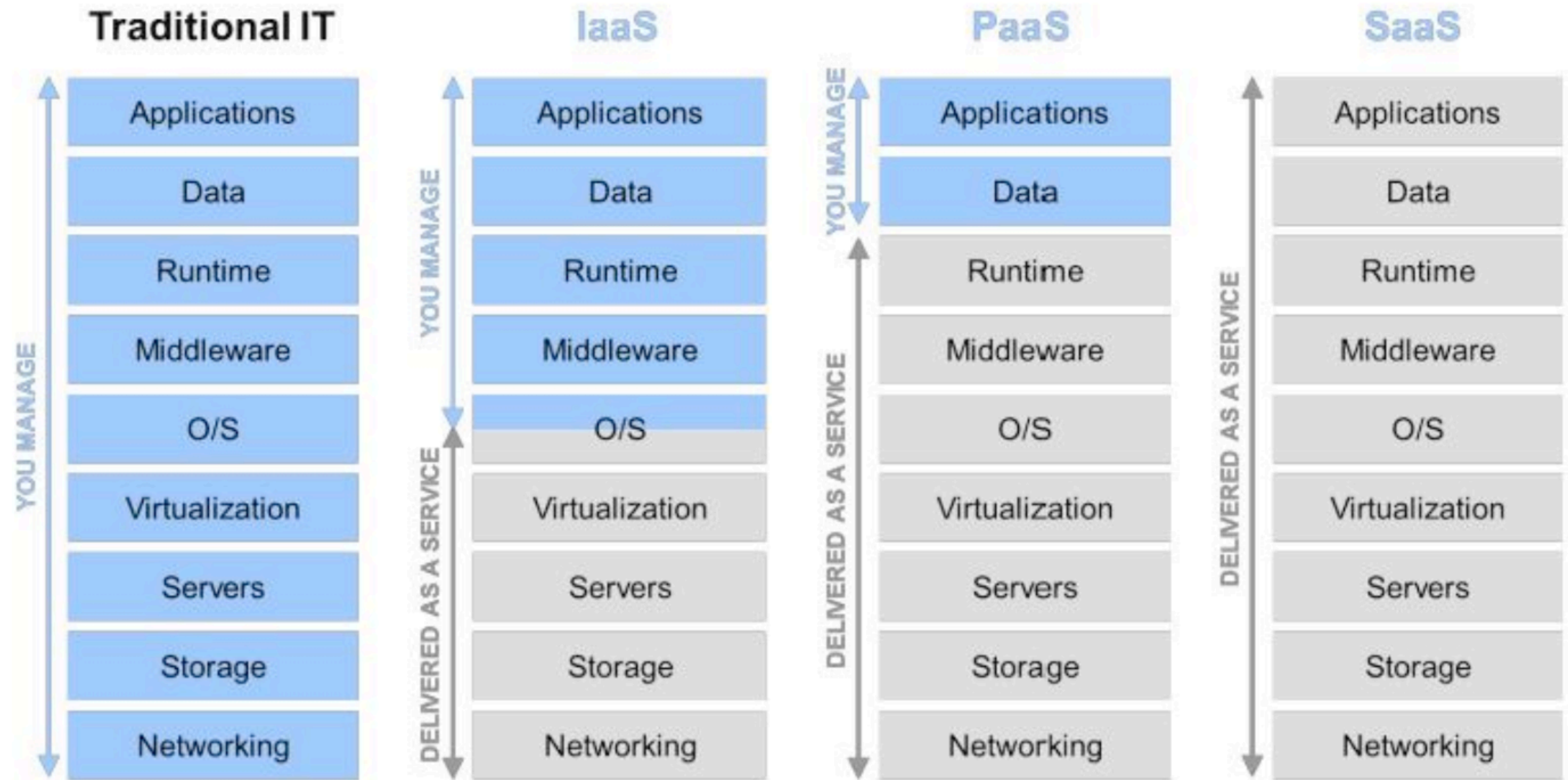
Ciclo de vida do desenvolvimento do software



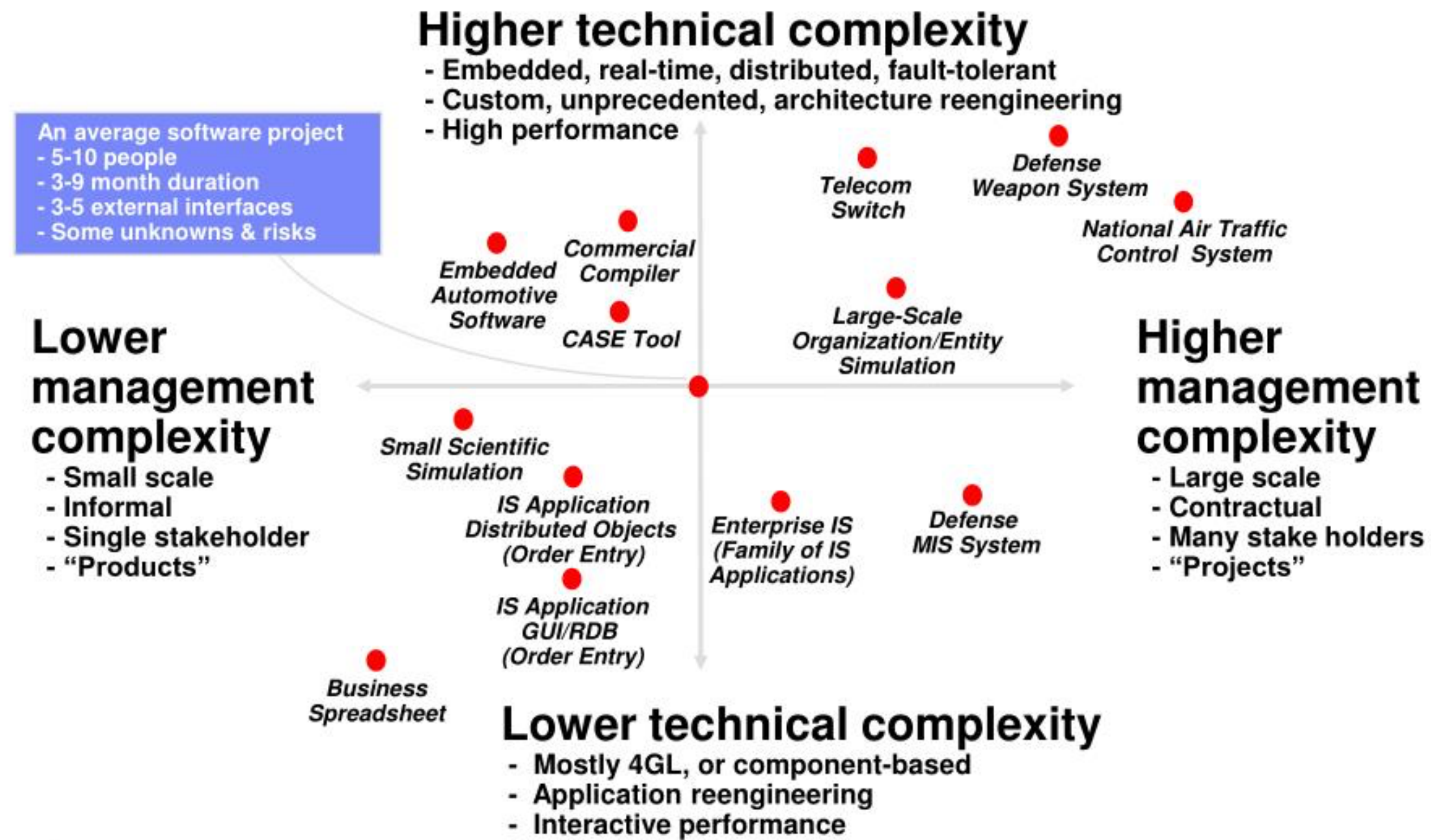
Processos de software







Dimensions of software complexity



Royce

7

© 2005 IBM Corporation

Exercícios

1. Indique os atributos necessários de um bom software.
2. Indique e explique as fases do ciclo de desenvolvimento de software.

Engenharia de Software

Introdução à engenharia de software