



Engenharia de Software

Guia de Estudo

Introdução à engenharia de software

Objectivos de aprendizagem

- Definir o que se entende por software
- Definir o que se entende por engenharia de software
- Compreender as motivações para uma abordagem sistemática e quantificável no desenvolvimento de software
- Compreender a natureza e efeito da complexidade e da mudança rápida no âmbito do desenvolvimento de software

Bibliografia

- Pressman, cap. 1
- Schach, cap. 1

Processos de desenvolvimento de software

Objectivos de aprendizagem – Parte 1

- Compreender o conceito de processo de desenvolvimento de software
- Compreender os conceitos de maturidade e capacidade no desenvolvimento de software
- Definir as principais actividades de um processo de desenvolvimento de software
- Compreender os princípios base de um processo de desenvolvimento linear

Objectivos de aprendizagem – Parte 2

- Compreender a relação entre conhecimento e risco no desenvolvimento de software
- Compreender o conceito de desenvolvimento cíclico, incremental e iterativo
- Conhecer os principais aspectos do processo RUP
- Definir a relação entre tipos de aplicações e características do processo de desenvolvimento a utilizar, nomeadamente, no que se refere ao nível de formalidade e ao carácter iterativo

Objectivos de aprendizagem – Parte 3

- Compreender o conceito de desenvolvimento ágil
- Conhecer os princípios base de desenvolvimento ágil
- Compreender a organização de um processo de desenvolvimento de software ágil
- Conhecer técnicas de desenvolvimento de software associadas a metodologias ágeis

Bibliografia

- Pressman, cap. 2, 3
- Larman, cap. 2

Análise de requisitos

Objectivos de aprendizagem

- Compreender o conceito de análise de requisitos
- Compreender o enquadramento da análise de requisitos no contexto de um processo de desenvolvimento de software
- Definir os elementos e actividades envolvidas na elaboração de uma especificação de requisitos
- Definir os conceitos de caso de utilização e de modelo de casos de utilização
- Compreender a forma de elaborar um modelo de casos de utilização
- Compreender os diagramas de casos de utilização da linguagem UML

Bibliografia

- Pressman, cap. 7, 8
- Larman, cap. 5, 6, 7

Introdução ao desenvolvimento de sistemas

Objectivos de aprendizagem

- Definir a noção de sistema e as principais características de um sistema
- Definir as principais perspectivas de descrição de um sistema computacional
- Compreender as noções de complexidade e de entropia no âmbito do desenvolvimento de software
- Compreender os conceitos de complexidade organizada e desorganizada e a sua relação com conhecimento e organização de um sistema

Bibliografia

- Booch, cap. 1

Modelos computacionais

Objectivos de aprendizagem

- Definir o problema da representação na especificação de software
- Compreender a especificação de software a diferentes níveis de representação
- Definir o conceito de modelo em engenharia de software e as principais características que deve apresentar
- Compreender a abordagem de modelação orientada a objectos
- Compreender a relação entre diferentes abordagens de modelação

Bibliografia

- Booch, cap. 2, 3

Linguagem UML

Objectivos de aprendizagem

- Compreender o âmbito, motivação e áreas de aplicação da linguagem UML
- Definir os conceitos base e os mecanismos de extensão da linguagem UML
- Definir os diferentes tipos de diagramas e a sua relação com diferentes perspectivas de modelação, nomeadamente, diagramas de classes, de objectos, de estrutura composta, de componentes, de implantação e de suporte à organização dos modelos (os restantes diagramas são estudados no contexto dos capítulos respectivos)

Bibliografia

- Seidl, cap. 1, 2, 4, 9
- UML v2.5.1

Arquitectura de software

Objectivos de aprendizagem

- Definir o conceito de arquitectura de software
- Definir diferentes níveis de arquitectura de software
- Compreender o enquadramento da actividade de projecto de arquitectura de software
- Definir as principais métricas e princípios de arquitectura de software
- Compreender o conceito de padrão de arquitectura de software e a sua concretização a diferentes níveis de arquitectura
- Compreender os conceitos de arquitectura orientada ao domínio e arquitectura multi-camada
- Compreender o conceito de arquitectura detalhada
- Compreender os principais princípios de arquitectura orientada a objectos

Bibliografia

- Pressman, cap. 9-11, 14,
- Larman, cap. 8, 9

Projecto de arquitectura de software

Objectivos de aprendizagem

- Definir a actividade de projecto de arquitectura de software
- Compreender o conceito de modelo de análise
- Compreender o conceito de modelo de projecto (*design*)
- Compreender a forma de aplicar um processo de desenvolvimento iterativo no projecto de arquitectura de software

Bibliografia

- Larman, cap. 1, 8, 9

Modelo de domínio

Objectivos de aprendizagem

- Definir o conceito de modelo de domínio
- Identificar classes conceptuais relativas à especificação de requisitos
- Distinguir entre classes e atributos de domínio relevantes e não relevantes
- Compreender as vantagens da definição de um modelo de domínio

Bibliografia

- Larman, cap. 10 - 12, 25 - 27

Modelo de interacção

Objectivos de aprendizagem

- Definir o conceito de modelo de interacção
- Compreender o suporte da linguagem UML para definição de modelos de interacção
- Compreender os diagramas de sequência, de comunicação, os diagramas temporais e os diagramas de enquadramento de interacção da linguagem UML

Bibliografia

- Larman, cap. 15
- Seidl, cap. 6
- UML v2.5.1

Modelo de dinâmica

Objectivos de aprendizagem

- Compreender o conceito de dinâmica de um sistema computacional
- Compreender os conceitos de estado e de espaço de estados
- Definir o conceito de máquina de estados
- Compreender os diagramas de transição de estados da linguagem UML
- Compreender o conceito de máquina de estados hierárquica, bem como o suporte da linguagem UML para a sua modelação
- Compreender os diagramas de actividade da linguagem UML
- Compreender as diferentes vertentes de utilização de diagramas de actividade

Bibliografia

- Sipser, cap. 1.1
- Larman, cap. 29
- Seidl, cap. 5, 7

Implementação

Objectivos de aprendizagem

- Compreender os principais níveis de especificação de software e a sua relação com a actividade de implementação
- Definir o processo de implementação
- Definir a conversão entre modelo e implementação
- Compreender a implementação da estrutura e da dinâmica de um sistema de software
- Compreender a implementação de máquinas de estados

Bibliografia

- Schach, cap. 15
- Larman, cap. 20

Refactorização e anti-padrões de software

Objectivos de aprendizagem

- Compreender a motivação e âmbito do conceito de refactorização de software
- Compreender em que circunstâncias e de que forma deve ser realizada a refactorização de software
- Compreender o conceito de anti-padrão de software
- Conhecer a relação entre refactorização e anti-padrões de software
- Conhecer indicadores de problemas numa arquitectura de software
- Conhecer alguns dos principais anti-padrões de software

Bibliografia

- Pressman, cap. 14

Verificação e teste

Objectivos de aprendizagem

- Compreender a motivação e enquadramento das actividades de verificação e teste no âmbito de um processo de desenvolvimento de software
- Compreender os conceitos de revisão e inspecção de software
- Definir o conceito de teste de software e os princípios de teste de software
- Definir níveis, tipos e técnicas de teste de software
- Definir o processo de teste de software
- Compreender o conceito de desenvolvimento guiado por testes

Bibliografia

- Pressman, cap. 19 - 21