

# Análise e Modelagem de Sistemas

## Introdução à Engenharia de Software e à Análise de Sistemas

Profª. Drª. Iolanda Cláudia Sanches Catarino

1

- Unidade de Ensino: 1.
- Competência da Unidade: Conhecer os fundamentos da Engenharia de Software e sua evolução, a classificação dos Sistemas de Informação, os modelos de processo de softwares e suas características.
- Resumo: Estudo sobre os fundamentos da Engenharia de Software e sua evolução, contextualização sobre os Sistemas de Informação, e o estudo sobre os modelos de processo de softwares e suas características.
- Palavras-chave: Engenharia de Software, Sistemas de Informação, Modelos de Processo de Software, Desenvolvimento Ágil.
- Título da Teleaula: Introdução à Engenharia de Software e a Análise de Sistemas
- Teleaula nº: 01

2

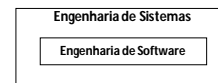
## Conceitos

# Fundamentos da Engenharia de Software

3

## Fundamentos da Engenharia de Software

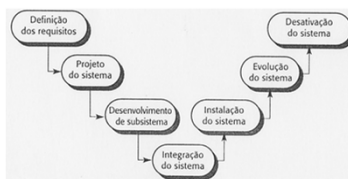
- A **Engenharia de Sistemas** abrange todos os aspectos relacionados ao desenvolvimento de sistemas com base em computadores, incluindo hardware, software e engenharia de processos (SOMMERVILLE, 2011).
- A **Engenharia de Software** é uma parte da Engenharia de Sistemas que se ocupa de todos os aspectos da produção de software (SOMMERVILLE, 2011).



4

## Fundamentos da Engenharia de Software

- Processo da Engenharia de Sistemas:

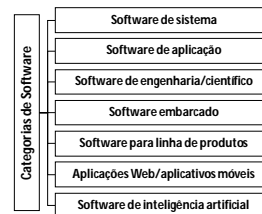


Fonte: Sommerville, 2011.

5

## Fundamentos da Engenharia de Software

- Campos de aplicação de software (PRESSMAN & MAXIM, 2016):



6

## Fundamentos da Engenharia de Software



- **Evolução de Software:**
  - WebApps.
  - Aplicativos móveis.

Fonte: PRESSMAN & MAXIM (2016, p.10).

7

# Fundamentos da Engenharia de Software

- Para desenvolver um Sistema de Informação (SI), o Gerente de Projetos de TI e sua equipe, entre eles o **Analista de Sistemas**, deve definir uma **metodologia de desenvolvimento** de sistemas que contemple procedimentos, um ou mais métodos com suas técnicas de modelagem e as tecnologias a serem adotados no desenvolvimento do sistema, visando a qualidade do software.

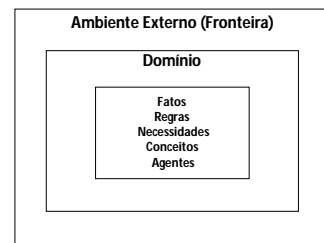
8

# Fundamentos da Engenharia de Software

- **Domínio do Sistema:** engloba o contexto para o qual será provida uma solução.
- O processo de reconhecimento do domínio caracteriza pela definição de sua **fronteira**.
- O conteúdo de um domínio compreende fatos, regras, necessidades, conceitos, agentes etc, ou seja, conhecer e definir o **contexto** do Sistema.

9

# Fundamentos da Engenharia de Software



10

## Fundamentos da Engenharia de Software

- O profissional **Analista de Sistemas** precisa ter habilidades e competências para:
  - Analisar, projetar e desenvolver sistemas computacionais principalmente SI.
  - Definir o uso adequado de Tecnologias de Informação e Comunicação (TIC), conforme o domínio dos projetos.
  - Visualizar o sistema computacional de várias perspectivas, abstraindo soluções lógicas e físicas satisfatoriamente.
  - Trabalhar com pessoas em equipe, mediar conflitos de ideias e gerenciar situações que exigem tomadas de decisão.

11

## Conceitos

# Sistemas de Informação

12

## Sistemas de Informação

- Os **Sistemas de Informação (SI)** podem ser definidos tecnicamente como um conjunto de componentes **inter-relacionados** que **coletam** (ou recuperam), **processam**, **armazenam** e **distribuem** informações destinadas a apoiar a tomada de decisões, a coordenação e o controle em uma organização (LAUDON & LAUDON, 2014).

13

## Sistemas de Informação

- Conceitos fundamentais – evolução da informação:**



14

## Sistemas de Informação

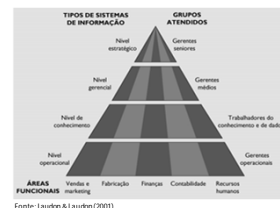
- Integração das dimensões de um Sistema de Informação:**
  - O principal objetivo de um SI é permitir que através de uma melhor utilização dos dados, possam ser definidas as informações que possuem **maior relevância para o processo organizacional**.



15

## Sistemas de Informação

- Níveis organizacionais x Tipos de Sistemas de Informação:**



16

## Sistemas de Informação

- Classificação de Sistemas Empresariais em quatro grupos básicos e funcionais:**
  - Sistemas de Processamento de Transações (SPT).
  - Sistemas de Informação Gerenciais (SIG).
  - Sistemas de Apoio à Decisão (SAD).
  - Sistemas de Apoio ao Executivo (SAE).

17

## Sistemas de Informação

- Sistemas de Processamento de Transações (SPT) - anos 1960:**
  - Transações rotineiras: Registro de pedidos de venda; Sistemas de reservas de hotel; Folha de pagamento; Registro de funcionários e expedição.
- Sistemas de Informações Gerenciais (SIG) - anos 1970:**
  - Informações resumidas.
  - Operações básicas da empresa.
  - Relatórios de desempenho.

18

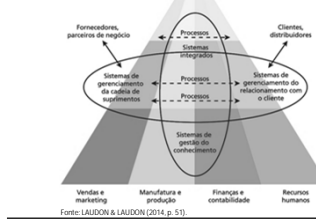
## Sistemas de Informação

- **Sistemas de Apoio à Decisão (SAD) - anos 1980:**
  - Microcomputadores nos ambientes empresariais.
  - Suporte para os executivos, diretores e gerentes.
  - Decisões pouco usuais – problemas esporádicos.
- **Sistemas de Apoio ao Executivo (SAE) - anos 1980/90:**
  - Suporte às questões estratégicas.
  - Focado o planejamento de longo prazo.
  - Gráficos personalizados.
  - Análises de tendências.

19

## Sistemas de Informação

### Áreas funcionais:

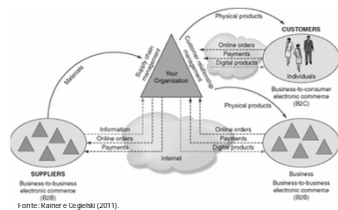


- Os **Aplicativos Integrados (sistemas empresariais)** ajudam as empresas a se tornarem mais produtivas, coordenando seus processos de negócios de forma integrada, a fim de melhorar o desempenho da organização como um todo.

20

## Sistemas de Informação

### Organização moderna baseada na Web:



21

## Conceitos

## O Processo de Software

22

## O Processo de Software

- Na concepção de Pressman e Maxim (2016), a **Engenharia de Software** abrange um conjunto de três elementos: processo, métodos e ferramentas. A base da engenharia de software é camada de processos.
- Camadas da Engenharia de Software:



Fonte: PRESSMAN & MAXIM (2016, p.16).

23

## O Processo de Software

- **Processo:** “processo de engenharia de software é um conjunto de atividades, ações e tarefas realizadas na criação de algum artefato. O processo constitui a base para o gerenciamento de projetos de software [...] e o processo define uma metodologia que deve ser estabelecida para a entrega efetiva de tecnologia de engenharia de software”. (PRESSMAN & MAXIM, 2016, p.15-16).
- Refere-se às decisões que serão tomadas quanto ao planejamento do projeto, a escolha do método com as técnicas de modelagem que serão especificadas e demais padrões adotados, no desenvolvimento do software.

24

## O Processo de Software

- **Método:** proporciona os detalhes de “como fazer” para construir o software. Envolve um amplo conjunto de atividades que incluem: planejamento, análise de requisitos, análise e projeto, implementação e testes.
- Para Sommerville (2011), um método é uma abordagem estruturada para o desenvolvimento de software, facilitando a produção de software com qualidade e uma boa relação custo-benefício.

25

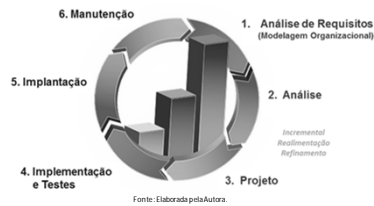
## O Processo de Software

- **Ferramentas:** proporciona apoio automatizado aos métodos de desenvolvimento de software, como por exemplo, as ferramentas CASE (*Computer Assisted Software Engineering* – Engenharia de Software Assistida por Computador) de modelagem, de banco de dados e de linguagens de programação.

26

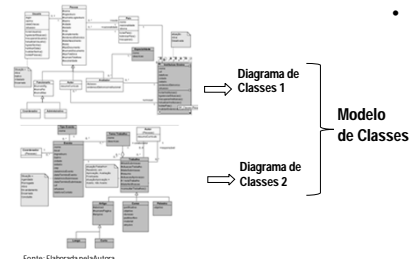
## O Processo de Software

- **Processo Básico de Desenvolvimento de Software – Ciclo de Vida:**



27

## O Processo de Software



- Todos os métodos apresentam **técnicas de modelagem** que possam ser representadas graficamente, na maioria, para especificar a modelagem do sistema.

28

## O Processo de Software

- “Um **modelo de processo** fornece um guia específico para o trabalho de engenharia de software. Ele define o fluxo de todas as atividades, ações e tarefas, o grau de iteração, os artefatos e a organização do trabalho a ser feito”. (PRESSMAN & MAXIM, 2016, p. 40).
  - Quem realiza?
  - Por que é importante?
  - Quais são as etapas envolvidas?
  - Qual é o artefato?
  - Como garantir que o trabalho foi realizado corretamente?

29

Resolução da SP

## Evolução de Software - Reflexão

30

## Interação

# Envie suas dúvidas!

31

## Conceitos

# Modelos de Processo Prescritivo

32

## Modelos de Processo Prescritivo

- Um **modelo de processo prescritivo** (modelo de processo tradicional) concentra-se em estruturar e ordenar o desenvolvimento de software, sendo que as atividades e tarefas ocorrem sequencialmente, com diretrizes de progresso definidas (PRESSMAN & MAXIM, 2016).
- Cada modelo de processo prescreve um fluxo de processo (fluxo de trabalho), ou seja, a forma pela qual os elementos do processo estão relacionados.

33

## Modelos de Processo Prescritivo

- Modelo Cascata:** “algumas vezes chamado ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, começando com a especificação dos requisitos do cliente, avançando pelas fases de planejamento, modelagem, construção e disponibilização, e culminando no suporte contínuo do software concluído” (PRESSMAN & MAXIM, 2016, p.42).
- O modelo cascata é o paradigma mais antigo da engenharia de software.

34

## Modelos de Processo Prescritivo

- O **Modelo Cascata:**



Fonte: PRESSMAN & MAXIM (2016, p.42).

35

## Modelos de Processo Prescritivo

- Modelo de Processo Incremental:** combina os fluxos de processo linear e paralelo dos elementos, aplicando sequências lineares de forma escalonada, à medida que o tempo vai avançando, sendo que cada sequência linear produz “incrementos” entregáveis do software (PRESSMAN & MAXIM, 2016).
- O modelo incremental libera uma série de versões, denominadas incrementos, que oferecem, progressivamente, maior funcionalidade ao cliente à medida que cada incremento é entregue.

36

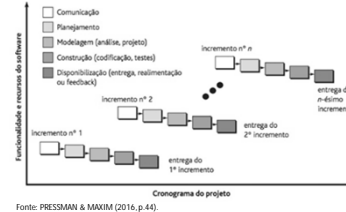
## Modelos de Processo Prescritivo

- **Modelos de Processo Evolucionário:** contemplam um modelo de processo que tenha sido projetado especificamente para desenvolver um produto que cresce e muda. Os modelos de processo evolucionário produzem uma versão cada vez mais completa do software a cada iteração (PRESSMAN & MAXIM, 2016).
- **Tipos:**
  - Prototipação.
  - Modelo Espiral

37

## Modelos de Processo Prescritivo

- **O Modelo de Processo Evolucionário:**



38

## Modelos de Processo Prescritivo

- **Prototipação:** modelo que auxilia os envolvidos a compreenderem melhor o que está para ser construído quando os requisitos estão obscuros (PRESSMAN & MAXIM, 2016).
- Começa com a comunicação e identifica-se os requisitos já conhecidos.
- Uma iteração de prototipação é planejada rapidamente e ocorre a modelagem na forma de projeto rápido.
- O projeto rápido leva à construção de um protótipo.
- O protótipo é entregue e avaliado pelos envolvidos, os quais fornecem feedback que é usado para refinar ainda mais os requisitos.

39

## Modelos de Processo Prescritivo

- **O Modelo de Processo Evolucionário - Prototipação:**



40

## Modelos de Processo Prescritivo

- **Modelo Espiral:** "Originalmente proposto por Barry Boehm, o modelo espiral é um modelo de processo de software evolucionários que une a natureza iterativa da prototipação aos aspectos sistemáticos e controlados do modelo cascata. Tem potencial para o rápido desenvolvimento de versões cada vez mais completas do software" (PRESSMAN & MAXIM, 2016, p.47).
- Com o modelo espiral, o software é desenvolvido em uma série de versões evolucionárias progressivamente com versões cada vez mais sofisticadas do software.

41

## Modelos de Processo Prescritivo

- **O Modelo de Processo Evolucionário - Espiral:**



42

## Modelos de Processo Prescritivo

- **Modelo Concorrente:** "por vezes chamado de engenharia concorrente, possibilita à equipe de software representar elementos concorrentes e iterativos de qualquer um dos modelos de processo" (PRESSMAN & MAXIM, 2016, p.49).
- A modelagem concorrente se aplica a todos os tipos de desenvolvimento de software e fornece uma imagem precisa do estado atual de um projeto. Em vez de limitar as atividades, ações e tarefas da engenharia de software a uma sequência de eventos, ela define uma rede de processos.

43

## Modelos de Processo Prescritivo

- **Um elemento do Modelo de Processo Concorrente:**



Fonte: PRESSMAN &amp; MAXIM (2016, p.51)

44

## Conceitos

## Modelos de Processo Especializado e o Processo Unificado

45

## Modelos de Processo Especializado

- Os **modelos de processo especializado** incluem muitas das características de um ou mais dos modelos tradicionais, sendo adotado quando se opta por uma abordagem de engenharia de software especializada ou definida de forma restrita (PRESSMAN & MAXIM, 2016).

46

## Modelos de Processo Especializado

- **Desenvolvimento Baseado em Componentes:** compreende aplicações de componentes de software previamente empacotados, incorporando muitas das características do modelo espiral (PRESSMAN & MAXIM, 2016, p.52).
- É evolucionário por natureza, demandando uma abordagem iterativa para a criação de software.
- As atividades de modelagem e construção começam com a identificação de componentes candidatos.

47

## Modelos de Processo Especializado

- **Etapas do Desenvolvimento Baseado em Componentes** (PRESSMAN & MAXIM, 2016, p.53):

  1. Produtos baseados em componentes disponíveis são pesquisados e avaliados para o campo de aplicação em questão.
  2. Itens de integração de componentes são considerados.
  3. Uma arquitetura de software é projetada para acomodar os componentes.
  4. Os componentes são integrados à arquitetura.
  5. Testes completos são realizados para garantir a funcionalidade adequada.

48



### Modelos de Processo Especializado

- **Modelo de Métodos Formais:** “inclui um conjunto de atividades que conduzem à especificação matemática formal do software. Eles permitem especificar, desenvolver e verificar um sistema baseado em computador pela aplicação de uma notação matemática rigorosa” (PRESSMAN & MAXIM, 2016, p.53).

49

### Modelos de Processo Especializado

- **Desenvolvimento de Software Orientado a Aspectos:** define “aspectos” que representam preocupações do cliente que transcendem várias funções, recursos e informações do sistema (PRESSMAN & MAXIM, 2016, p.54).
  - É um paradigma de engenharia de software relativamente novo que oferece uma abordagem metodológica e de processos para definir, especificar, projetar e construir aspectos – “mecanismos além das sub-rotinas e herança para localizar a expressão de uma preocupação cruzada.

50

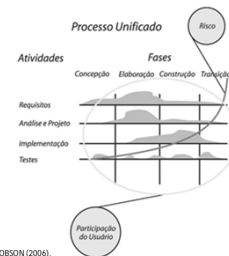
### Processo Unificado

- **Processo Unificado:** “é um processo de software dirigido a casos de uso, centrado na arquitetura, iterativo e incremental” (PRESSMAN & MAXIM, 2016, p.55).
- Sugere um fluxo de processo iterativo e incremental, proporcionando a sensação evolucionária que é essencial no desenvolvimento de software moderno.
- Um fluxo de trabalho de engenharia de software é distribuído ao longo de todas as fases do Processo Unificado.

51

### Processo Unificado

- **O Processo Unificado:**



Fonte: ROODH, RUMBAUGH e JACOBSON (2006).

52

Conceitos

## Introdução ao Desenvolvimento Ágil

53

### Introdução ao Desenvolvimento Ágil

- “Em 2001, Kent Beck e outros 16 renomados desenvolvedores, autores e consultores da área de software (batizados de *Agile Alliance* – Aliança dos Ágeis) assinaram o **Manifesto para o Desenvolvimento Ágil de Software** (Manifesto for *Agile Software Development*)” (PRESSMAN & MAXIM, 2016, p.66).

54

## Introdução ao Desenvolvimento Ágil

- **O Manifesto para o Desenvolvimento Ágil de Software, declarava** (PRESSMAN & MAXIM, 2016, p.66):

*Ao desenvolver e ajudar outros a desenvolver software, desvendamos formas melhores de desenvolvimento. Por meio deste trabalho passamos a valorizar:*

- *Indivíduos e interações acima de processos e ferramentas.*
- *Software operacional acima de documentação completa.*
- *Colaboração dos clientes acima de negociação contratual.*
- *Respostas a mudanças acima de seguir um plano.*

55

## Introdução ao Desenvolvimento Ágil

- **Características** (PRESSMAN & MAXIM, 2016, p.66):

- A engenharia de software ágil combina filosofia com um conjunto de princípios de desenvolvimento.
- A filosofia defende a satisfação do cliente e a entrega incremental antecipada.
- Equipes de projeto pequenas e altamente motivadas.
- Comunicação ativa e contínua entre desenvolvedores e clientes.
- Métodos informais.
- Artefatos de engenharia de software mínimos.
- Simplicidade no desenvolvimento geral.
- Entrega mais do que a análise e o projeto.

56

## Introdução ao Desenvolvimento Ágil

- **Doze Princípios** (PRESSMAN & MAXIM, 2016, p.70-71):

1. A maior prioridade é satisfazer o cliente com entrega antecipada e contínua de software funcionando.
2. Aceite bem os pedidos de alterações, mesmo com o desenvolvimento antecipado. Os processos ágeis se aproveitam das mudanças para a vantagem competitiva do cliente.
3. Entregue software em funcionamento frequentemente, de algumas semanas a alguns meses, dando preferência a intervalos mais curtos.
4. O pessoal do comercial e os desenvolvedores devem trabalhar em conjunto diariamente ao longo de todo o projeto.

57

## Introdução ao Desenvolvimento Ágil

5. Construa projetos em torno de pessoas motivadas. Dê a elas o ambiente e o apoio necessários e acredite que elas farão o trabalho corretamente.
6. O método mais eficiente e efetivo de transmitir informações para e dentro de uma equipe de desenvolvimento é uma conversa aberta, presencial.
7. Software em funcionamento é a principal medida de progresso.
8. Os processos ágeis promovem desenvolvimento sustentável. Proponentes, desenvolvedores e usuários devem estar aptos a manter um ritmo constante indefinidamente.
9. Atenção contínua para com a excelência técnica e para com bons projetos aumenta a agilidade.

58

## Introdução ao Desenvolvimento Ágil

10. Simplicidade – a arte de maximizar o volume de trabalho não realizado – é essencial.
11. As melhores arquiteturas, requisitos e projetos surgem de equipes auto-organizadas.
12. Em intervalos regulares, a equipe se avalia para ver como pode se tornar mais eficiente, então, sintoniza e ajusta seu comportamento de acordo.

59

## Introdução ao Desenvolvimento Ágil

- **Principais Métodos de Processos Ágeis:**

- *eXtreme Programming (XP).*
- *Scrum.*
- *Dynamic Systems Development Method (DSDM - Método de Desenvolvimento de Sistemas Dinâmicos).*
- *Modelagem Ágil (AM).*
- *Processo Unificado Ágil (AUP).*
- *Feature Driven Development (FDD).*
- *Microsoft Solutions Framework (MSF).*

60

Resolução da SP

## Modelos de Processo de Software - Reflexão

61

Interação

## Envie suas dúvidas!

62

### Referências e Bibliografia

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3ed. Rio de Janeiro: Elsevier, 2014.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **The unified modeling language: user guide**. Massachusetts: Longman, 2000.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. 2ed. Rio de Janeiro: Elsevier, 2006.

GUEDES, Gilleanes T. A.. **UML 2: uma abordagem prática**. 3ed. São Paulo: Novatec, 2018.

KIRCHOFF, Egon. **BPMN em exemplos**: aprenda como modelar processos de negócio. Kirchoff, 2015.

LAUDON, K. C.; LAUDON, J. Price. **Gerenciamento de sistemas de informação**. 3 ed. LTC: Rio de Janeiro, 2001.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de informação gerenciais**. 11 ed. São Paulo: Pearson Prentice Hall, 2014.

63

### Referências e Bibliografia

PRESSMAN, R.; MAXIM, B.. **Engenharia de software**: uma abordagem profissional. 8 ed. Porto Alegre: AMGH, 2016.

RAINER, R. K.; CEGIELSKI, C. G. **Introdução a sistemas de informação**: apoiando e transformando negócios na era da mobilidade. 5ed. Rio de Janeiro: Elsevier, 2016.

RUMBAUGH, James et al.. **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Campus, 1994.

SOMMERVILLE, Ian. **Engenharia de software**. 9ed. São Paulo: Addison-Wesley, 2011.

VALLE, R.; OLIVEIRA, S. B. (Org.). **Análise e modelagem de processos de negócio**: foco na notação BPMN (Business Process Modeling Notation). São Paulo: Atlas, 2013.

64



65