



Engenharia de Software

ARA0097

Profª Otília C Nogueira

otilia.nogueira@estacio.br

Ementa

Ao final da disciplina, o aluno estará apto a

- interpretar os conceitos iniciais da engenharia de software, com base na necessidade da sua utilização, para melhor compreender o funcionamento de um software;
- analisar as etapas para de desenvolver um software, com base nas etapas definidas pela engenharia de software, para entregar um bom produto;
- distinguir os processos de desenvolvimento de software, com base nos seus métodos e procedimentos, para aprender a modelar e estruturar o desenvolvimento de sistemas;
- examinar o processo de criação de um sistema, com base nas boas práticas de engenharia de software, para garantir a qualidade do produto; e
- realizar o gerenciamento de configurações, com base nas principais técnicas utilizadas no mercado de trabalho, para manter o controle de versões de software.

Objetivos



- Interpretar os conceitos iniciais da engenharia de software, com base na necessidade da sua utilização, para melhor compreender o funcionamento de um software.
- Analisar as etapas para de desenvolver um software, com base nas etapas definidas pela engenharia de software, para entregar um bom produto.
- Distinguir os processos de desenvolvimento de software, com base nos seus métodos e procedimentos, para aprender a modelar e estruturar o desenvolvimento de sistemas.
- Examinar o processo de criação de um sistema, com base nas boas práticas de engenharia de software, para garantir a qualidade do produto.
- Realizar o gerenciamento de configurações, com base nas principais técnicas utilizadas no mercado de trabalho, para manter o controle de versões de software.

Conteúdo Programático

UNIDADE I - FUNDAMENTOS DE SOFTWARE E GERENCIAMENTO DE PROJETOS.

1. PROCESSO DE FABRICAÇÃO DO HARDWARE X PROCESSO DE DESENVOLVIMENTO DO SOFTWARE
2. ENGENHARIA DE SOFTWARE
3. O DESENVOLVIMENTO DE SOFTWARE COMO UM PROJETO E AS ATIVIDADES DE GERENCIAMENTO
4. PLANEJAMENTO E GERENCIAMENTO DE RISCOS



Conteúdo Programático

UNIDADE 2. FASES DO DESENVOLVIMENTO DE SOFTWARE

1. ENGENHARIA DE REQUISITOS E ANÁLISE DE SISTEMA
2. PROJETO DO SISTEMA
3. IMPLEMENTAÇÃO E TESTES
4. MANUTENÇÃO

Conteúdo Programático

Unidade III - MODELOS DE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE.

1. MODELOS DE PROCESSOS PRESCRITIVOS
2. O PROCESSO UNIFICADO
3. DESENVOLVIMENTO ÁGIL - EXTREME PROGRAMMING (XP)
4. DESENVOLVIMENTO ÁGIL - SCRUM E PROCESSO UNIFICADO ÁGIL

Conteúdo Programático

UNIDADE IV -QUALIDADE DE SOFTWARE (CRÉDITO DIGITAL)

1. QUALIDADE DE PROCESSO E DE PRODUTO DE SOFTWARE
2. PROCESSO DA GARANTIA DE QUALIDADE DE SOFTWARE
3. PLANEJAMENTO DA QUALIDADE E O CONTROLE DA QUALIDADE DE SOFTWARE
4. MEDIÇÕES E MÉTRICAS DO SOFTWARE

Conteúdo Programático

UNIDADE V - GERENCIAMENTO DE CONFIGURAÇÕES

1. FUNDAMENTOS DE GERENCIAMENTO DE CONFIGURAÇÕES
2. GERENCIAMENTO DE MUDANÇAS, VERSÕES E RELEASES
3. CONSTRUÇÃO DE SISTEMAS E FERRAMENTAS CASE

Procedimento de Avaliação

As avaliações serão presenciais e digitais, alinhadas à carga-horária da disciplina, divididas da seguinte forma:

- Avaliação 1 (AV1), Avaliação 2 (AV2), Avaliação Digital (AVD) e Avaliação 3 (AV3):
 - *AV1 - Contemplará os temas abordados na disciplina até a sua realização e será assim composta:
 - Prova individual com valor total de 7 (sete) pontos;
 - Atividades acadêmicas avaliativas com valor total de 3 (três) pontos, assim distribuídos:
 - Atividade da Aula 05 sobre casos de teste com valor de 1 ponto
 - Atividade da Aula 06 sobre testes de software com valor de 1 ponto
 - Atividade da Aula 08 sobre processo de desenvolvimento de software com valor 1 Ponto
 - A soma de todos os instrumentos que possam vir a compor o grau final da AV1 não poderá ultrapassar o grau máximo de 10 (dez) pontos.
 - *AV2 - Contemplará todos os temas abordados pela disciplina e será composta por uma prova teórica no formato PNI - Prova Nacional Integrada, que terá o seguinte formato:
 - Nas disciplinas que contém créditos teóricos e digitais: PNI de 0 a 10.
 - *AVD - Avaliação digital do(s) tema(s) / tópico(s) vinculado(s) ao crédito digital no valor total de 10 (dez) pontos.
 - *AV3 - Contemplará todos os temas abordados pela disciplina. Será composta por uma prova no formato PNI - Prova Nacional Integrada, com total de 10 pontos, substituirá a AV1 ou AV2 e não poderá ser utilizada como prova substituta para a AVD.
- Para aprovação na disciplina, o aluno deverá, ainda:
 - atingir resultado igual ou superior a 6,0, calculado a partir da média aritmética entre os graus das avaliações presenciais e digitais, sendo consideradas a nota da AVD ou AVDs e apenas as duas maiores notas obtidas dentre as três etapas de avaliação (AV1, AV2 e AV3). A média aritmética obtida será o grau final do aluno na disciplina;
 - obter grau igual ou superior a 4,0 em, pelo menos, duas das três avaliações presenciais e em uma das avaliações digitais (AVD ou AVDs);
 - frequentar, no mínimo, 75% das aulas ministradas.

Bibliografia Básica



- ANDRADE, Mayb. **Qualidade de Software**. 1ª Ed. Rio de Janeiro: SESES, 2015.
Disponível em: <https://repositoriov2.azurewebsites.net/api/objetos/efetuaDownload/405d3e91-3eef-4471-9ead-702cee3d2861>
- PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software**. Porto Alegre: AGMH, 2016.
Disponível em: [https://integrada\[minhabiblioteca\].com.br/#/books/9788580555349/](https://integrada[minhabiblioteca].com.br/#/books/9788580555349/)
- SOMMERVILLE, Ian. **Engenharia de Software**. 10ª Ed. São Paulo: Pearson Prentice Hall, 2011.
Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Loader/168127/pdf>

Calendário

fevereiro						
D	S	T	Q	Q	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

março						
D	S	T	Q	Q	S	S
			1	2	3	4
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

abril						
D	S	T	Q	Q	S	S
					1	2
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

maio						
D	S	T	Q	Q	S	S
			1			
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

junho						
D	S	T	Q	Q	S	S
			1	2	3	4
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

julho						
D	S	T	Q	Q	S	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

AV2

AV3



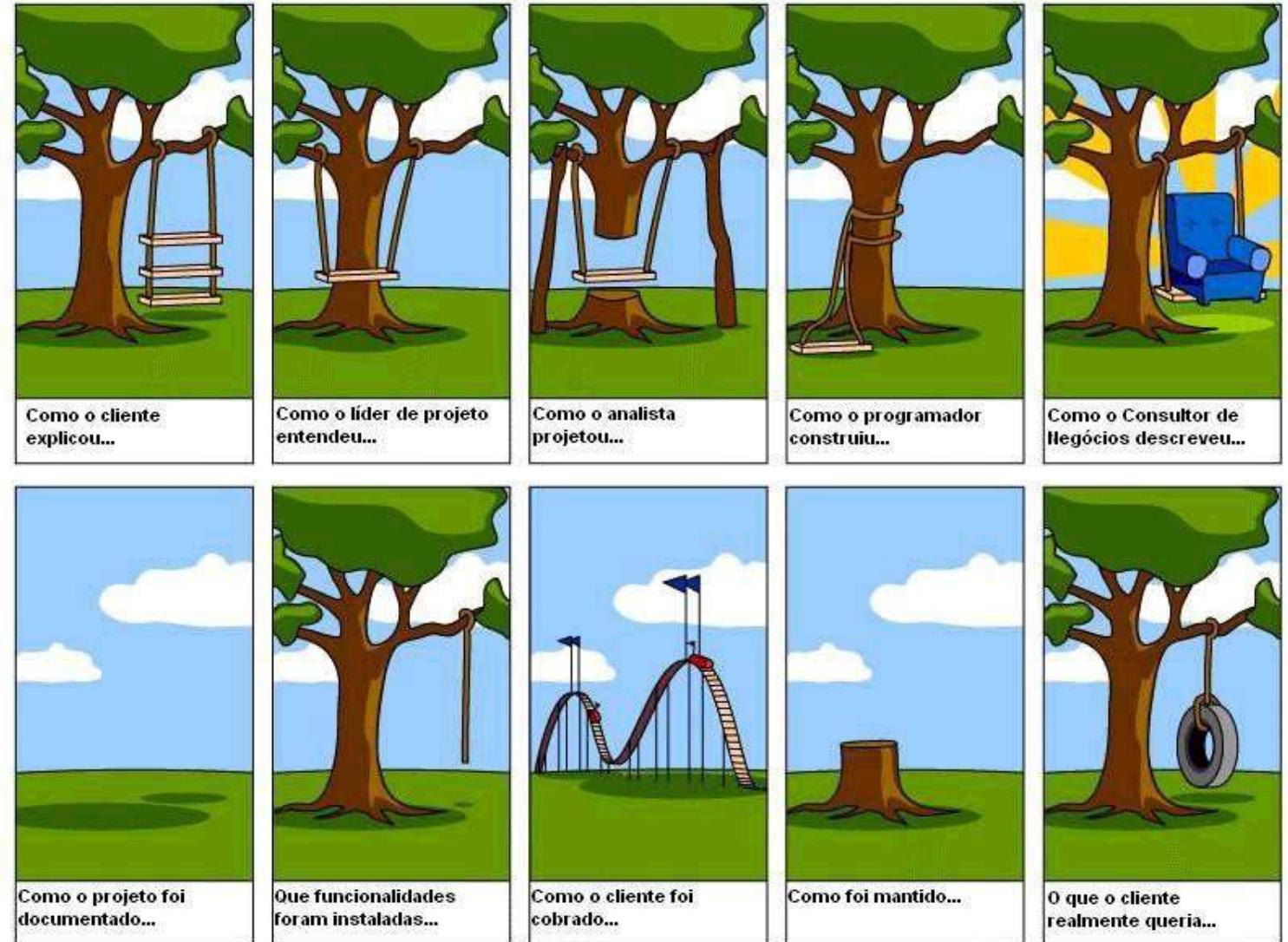
Fundamentos de Software e Gerenciamento de Projetos

Aula 01

Objetivos

- Reconhecer a necessidade do desenvolvimento de software ser com base em princípios da engenharia de software.

Imagen Clássica



O que é Engenharia de Software?

- Segundo **Pressman** (2006), um software é um conjunto composto por instruções de computador, estruturas de dados e documentos;
- Produtos de software podem ser:
 - **Genéricos** – desenvolvidos para serem vendidos para uma grande variedade de clientes, por exemplo, softwares para PC, tais como Excel e Word.
 - **Personalizados** – desenvolvidos para um único cliente de acordo com as suas especificações.

O que é Engenharia de Software?

- **Segundo o IEEE (1992),**
“Engenharia de software é a aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do software; isto é, a aplicação de engenharia ao software.”;
- **Para Bauer (1969) apud Pressman (2006),** a Engenharia de Software é “a criação e a utilização de sólidos princípios de engenharia a fim de obter softwares econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais”;

O que é Engenharia de Software?

- A Engenharia de Software almeja inserir as mesmas sistemáticas existentes em outras áreas da engenharia:
 - custos aceitáveis;
 - gerenciamento do processo de desenvolvimento;
 - garantia do trabalho em equipe e;
 - desenvolvimento de softwares com qualidade.

O que é Engenharia de Software?

- Engenheiros de software devem – dependendo do problema a ser resolvido, das restrições de desenvolvimento e dos recursos disponíveis – adotar uma abordagem sistemática e organizada para seu trabalho, além de usar ferramentas e técnicas apropriadas.



O que se estuda em engenharia de software?

- SWEBOK é precisamente documentar o corpo de conhecimento que caracteriza a área que hoje chamamos de Engenharia de Software.
- O SWEBOK define 12 áreas de conhecimento em Engenharia de Software:
 - 1) Engenharia de Requisitos
 - 2) Projeto de Software
 - 3) Construção de Software
 - 4) Testes de Software
 - 5) Manutenção de Software
 - 6) Gerência de Configuração
 - 7) Gerência de Projetos
 - 8) Processos de Software
 - 9) Modelos de Software
 - 10) Qualidade de Software
 - 11) Prática Profissional
 - 12) Aspectos Econômicos

Histórico

- **Década de 50**

- surgiram os primeiros softwares;
- pesquisas eram voltadas para o hardware;
- hardware disponível apenas nos centros de pesquisa;
- software desenvolvido sem utilizar técnicas de engenharia.

Histórico

- **Década de 60**

- surgiram os microprocessadores e o hardware deixou de representar um problema;
- software tornou-se o foco dos pesquisadores;
- organizações começaram a desenvolver grandes sistemas.



Commodore - CP/M-86 Version 1.0, Date: May 26, 1983

```
Commodore - CP/M-86 Version 1.0, Date: May 26, 1983
@dir
a: KEYDEF BAK : SYSTAT CMD : STAT    CMD : ASMB6 CID
a: GENCMD CMD : COPYDISK CMD : PIP     CMD : LOADER CID
a: GENDEF CMD : DDT86   CMD : SUBMIT  CID : TOR  CDR
a: TEST    A86 : CMM8050 LIB : REC    COM :
a: JOBTAB  BAK : ED    CMD : DEBLOCK BAK : 0
a: DEBLOCK DAT : CBIOS H86 : JOBTAB A86 : 1
a: KEYDEF A86 : CBIOS BAK
@]

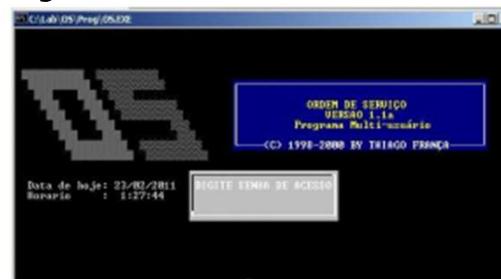
WHEN 3 MOVE WIN-DATA (1:5) TO S-ROAD-DESC
MOVE WIN-DATA (6:) TO S-ROAD-DESC
MODIFY EF-ROAD-HIERARCHY VALUE S-ROAD
MODIFY LB-ROAD-HIERARCHY-DESC
TITLE S-ROAD-HIERARCHY-DESC
WHEN 4 MOVE WIN-DATA (1:5) TO S-ROAD-TYPE
MOVE WIN-DATA (6:) TO S-ROAD-TYPE
MODIFY EF-ROAD-TYPE VALUE S-ROAD-TYPE
MODIFY LB-ROAD-TYPE-DESC VALUE S-ROAD-TYPE
MOVE DATA (1:5) TO S-ROAD-Quality
```

Surgimento da Eng. de Software

- as equipes de trabalho não tinham um modelo de como desenvolver;
- não havia documentação adequada do que estava sendo executado;
- “E agora: como dar manutenção em um sistema que não tem projeto?”

“O mundo de hoje não poderia viver sem o software”

Sommerville, 2011



Mas o que é software?



O que é software?

- Mais do que programas-fonte e executáveis: compreende também a documentação associada a ele, dados e configuração
- Tudo isso impacta a futura evolução e manutenção do software

Objetivos da Engenharia de Software

- Controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de qualidade desejados;
- Produtividade no desenvolvimento, operação e manutenção de software;
- Qualidade versus Produtividade;
- Permitir que profissionais tenham controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de qualidade desejados.

Características da Engenharia de Software

- A engenharia de software se refere a software (sistemas) desenvolvidos por grupos ao invés de indivíduos;
- Usa princípios de engenharia ao invés de arte, e....
- Inclui tanto aspectos técnicos quanto não técnicos.

O que é um software de qualidade?

- O software que satisfaz os requisitos solicitados pelo usuário. Deve ser fácil de manter, ter boa performance, ser confiável e fácil de usar.
- Alguns atributos de qualidade:
 - **Manutenibilidade:** O software deve evoluir para atender os requisitos que mudam.
 - **Eficiência:** O software não deve desperdiçar os recursos do sistema.
 - **Usabilidade:** O software deve ser fácil de usar pelos usuários para os quais ele foi projetado.

Qualidade de Software (ex. varejo)

- **Correto**

- A loja não pode deixar de cobrar por produtos comprados pelo consumidor.

- **Robusto e altamente disponível**

- A loja não pode parar de vender.

- **Eficiente**

- O consumidor não pode esperar.
- A empresa quer investir pouco em recursos computacionais (CPU, memória, rede).

Qualidade de Software (ex. varejo)

- **Amigável e fácil de usar**

- A empresa quer investir pouco em treinamento.

- **Altamente extensível e adaptável**

- A empresa tem sempre novos requisitos (para ontem!);
 - A empresa quer o software customizado do seu jeito (interface, teclado, idioma, moeda, etc).

- **Reusável**

- Várias empresas precisam usar partes de um mesmo sistema.

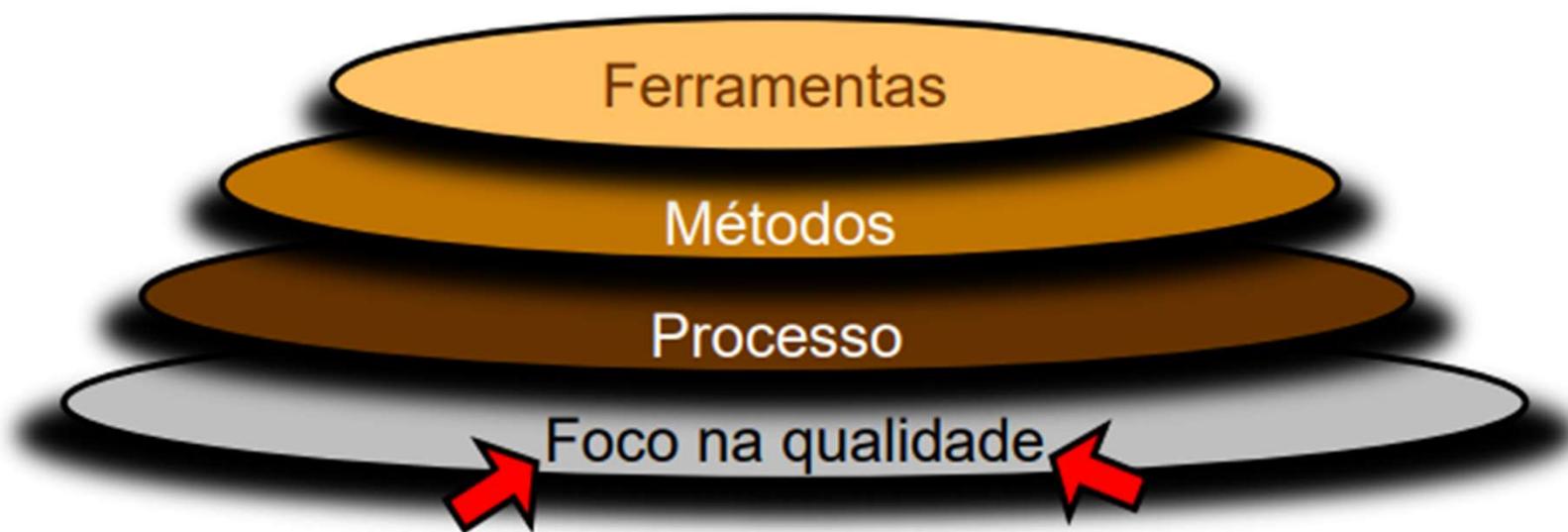
Qualidade de Software (ex. varejo)

- Aberto, compatível, de fácil integração com outros sistemas:
 - A empresa já tem controle de estoque, fidelização, etc.
- Portável e independente de plataforma (hw e sw)
 - A empresa opta por uma determinada plataforma.
- Baixo custo de instalação e atualização:
 - A empresa tem um grande número de PDVs.

Produtividade

- Custo de desenvolvimento reduzido:
 - A empresa consumidora quer investir pouco em software.
 - A empresa produtora tem que oferecer “software barato”.
- Tempo de desenvolvimento reduzido:
 - Suporte rápido às necessidades do mercado.

Camadas da Engenharia de Software



Camadas da Engenharia de Software

- **Processo (métodos + ferramentas)**

- une os métodos às ferramentas;
- define a sequência dos métodos que serão aplicados e;
- define as ferramentas que serão disponibilizadas.

- **Métodos (como fazer)**

- há diferentes métodos para as diferentes etapas do desenvolvimento;
- existem métodos para análise de requisitos, projeto, codificação, testes e manutenção.

- **Ferramentas (apoio automatizado)**

- CASE (Computer Aided Software Engineering);
- análise estruturada, análise essencial ou orientada a objetos;
- ferramentas de banco de dados;
- linguagens de programação.

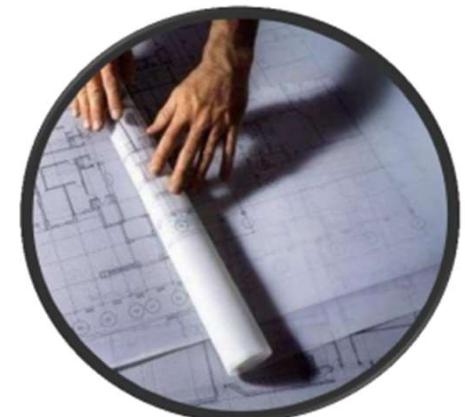
Estrutura de um Processo (1/4)

- Estrutura de um processo aplicável à maioria dos projetos de software, sejam eles de pequena, média ou alta complexidade:

1. **Comunicação**: Contempla alta comunicação e colaboração com o cliente/usuário e abrange o levantamento de requisitos.



2. **Planejamento**: Descreve as tarefas técnicas, os riscos prováveis, os recursos necessários, os produtos que serão produzidos e um cronograma.

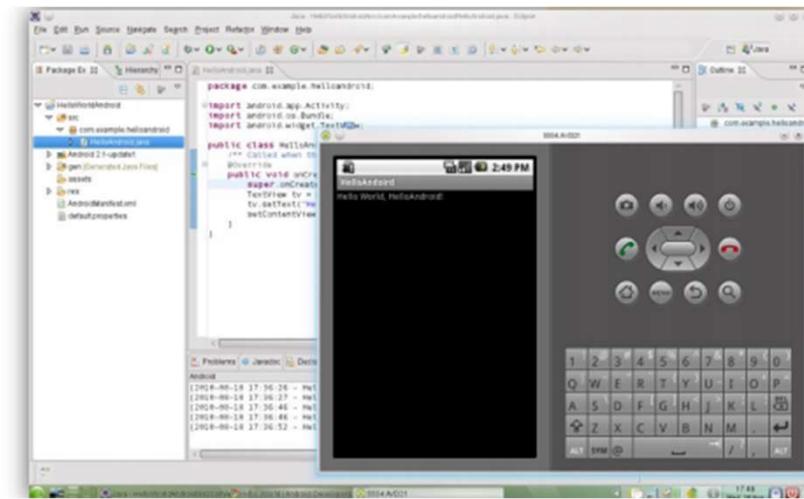
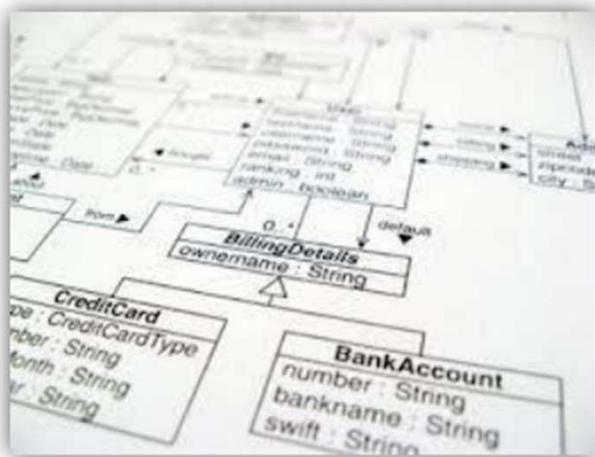


Estrutura de um Processo (2/4)

Estrutura de um processo aplicável à maioria dos projetos de software, sejam eles de pequena, média ou alta complexidade:

3. Modelagem: Constrói modelos que permitem ao desenvolvedor e ao cliente entender melhor os requisitos do software e o software que atenderá esses requisitos.

4. Construção: Contempla a criação dos códigos e a execução de testes.



Estrutura de um Processo (3/4)

Estrutura de um processo aplicável à maioria dos projetos de software, sejam eles de pequena, média ou alta complexidade:

5. **Implantação:** Avaliação e feedback do cliente quanto ao software desenvolvido



Estrutura de um Processo (4/4)

Um Processo de Software é o conjunto de atividades cuja meta é o desenvolvimento ou evolução de um software.

As atividades genéricas em todos os processos de software são:

- **Especificação**: o que o sistema deve fazer e suas restrições de desenvolvimento.
- **Desenvolvimento**: produção do sistema de software.
- **Validação**: verificação de que o software é o que o cliente deseja.
- **Evolução**: mudança do software em resposta às demandas de mudança.

