



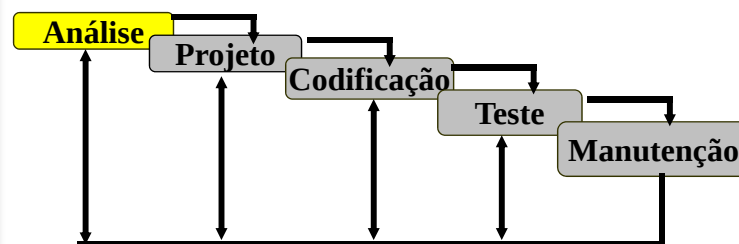
ENGENHARIA DE SOFTWARE

UNIDADE 3 – Fase de Análise (Etapas Iniciais)
(Aula 4 – Análise de Sistemas e Análise de Requisitos)

Prof. Ivan Nunes da Silva

1. Introdução

Revisão Introdutória



- Como vimos na aula anterior, dependendo do paradigma adotado, existem diferentes atividades a serem realizadas durante o desenvolvimento do software.
- A primeira fase (Análise) é dividida em duas, ou sejam:
 - Análise de Sistema.
 - Análise de Requisitos.

2. Análise de Sistemas

2.1. Introdução

- Esta é a primeira etapa do processo de engenharia de software.
- Em resumo, é nessa etapa que o engenheiro de software deve analisar:
 - Qual é o objetivo do software?
 - Para que o usuário vai utilizar esse software?
 - Que tipos de pessoas vão utilizar o software?
- Assim, nessa etapa deve-se analisar não apenas o software que será construído, mas todo o cenário envolvido na sua construção e destinação.
- Ou seja, deve-se analisar o ambiente de uso, usuários, mercado, concorrência, etc.



3

2. Análise de Sistemas

2.2. Identificação da Necessidade (1)

- O primeiro passo do processo de análise de sistema envolve a identificação da necessidade.
- É o ponto de partida na evolução de um sistema baseado em computador.
- Consiste em descobrir se realmente existe a necessidade de se produzir um software.
- Deve ser realizada para definir quais são as metas globais do sistema:
 - Qual a função desejada para o sistema?
 - Quais informações devem ser fornecidas para o sistema e produzidas pelo sistema?
 - Qual o desempenho desejado?
 - Influência: gastos com projeto.
 - Quais são as metas de qualidade e confiabilidade?
 - Influência: gasto com testes.



4

2. Análise de Sistemas

2.2. Identificação da Necessidade (2)

- Após definição de metas globais, passa-se para a avaliação das informações complementares:
 - Existe tecnologia para construir o sistema?
 - Quais recursos especiais de desenvolvimento e produção serão exigidos?
 - Quais limites foram estabelecidos para os custos e para os prazos?
- Se o produto for feito para produção em larga escala, as seguintes análises devem ser feitas:
 - Qual é o mercado em potencial para o produto?
 - Será um produto para perfis específicos? Engenharia/Comercial?
 - Como esse produto se compara com os produtos concorrentes?
 - Será mais eficiente? Mais amigável?
 - Que posição esse produto ocupa na linha de produtos da empresa?

5

2. Análise de Sistemas

2.3. Estudo da Viabilidade (1)

- Uma vez decidido que o software é necessário, ou que realmente será útil, realiza-se o estudo de viabilidade.
- Basicamente, consiste em verificar se é possível desenvolver o produto.
- Esta etapa é realizada pelos profissionais que desenvolverão o software.
- **Ou seja, de um modo geral: o sistema pode ser feito com o tempo, recursos e condições disponíveis?**
- Deverá se gastar mais tempo nesta etapa quanto mais complexo for o sistema e maior for o público-alvo:
 - Software de propósito específico.
 - Software de propósito geral.

6

2. Análise de Sistemas

2.3. Estudo da Viabilidade (2)

- Quatro principais áreas devem ser analisadas neste estudo:
 - **Viabilidade Econômica**
 - Avaliação do custo de desenvolvimento confrontada com a a renda de retorno derivada do sistema desenvolvido.
 - É a principal consideração para a maioria dos sistemas.
 - **Viabilidade Técnica**
 - Estudo da função desejada, do desempenho e das restrições que possam afetar a capacidade de se conseguir um sistema tecnicamente aceitável.
 - É a área mais difícil de ser avaliada nesta etapa.
 - Algumas considerações que devem ser levantadas:
 - Avaliar disponibilidade de pessoal competente para desenvolver o software.
 - Avaliar se há disponibilidade de recursos necessários (hardware e software) para a construção do sistema.
 - **Viabilidade Legal**
 - Verificação se o desenvolvimento do sistema resulta em alguma infração, violação ou responsabilidade legal. Alguns exemplos são os seguintes:
 - Software não indicado para menores de idade.
 - Software de aposta de jogos.
 - **Alternativas**
 - Avaliação de abordagens alternativas sobre o desenvolvimento do sistema. Exemplo:
 - Desenvolver um procedimento computacional que pode ser acoplado a um software já existente.

7

2. Análise de Sistemas

2.4. Análise Econômica (1)

- A partir do momento que decide que aparentemente o sistema é viável, deve ser realizada uma análise econômica ainda mais detalhada:
 - Pode ser realizada tanto pelos profissionais que estão desenvolvendo o software quanto pelo cliente que pretende comprar o software.
 - A análise custo-benefício define não apenas os objetivos tangíveis, mas também aqueles intangíveis.
 - Exemplos de benefícios intangíveis: maior satisfação do cliente, melhores decisões empresariais, etc.
- Alguns custos que devem ser avaliados para a confecção dos sistemas são os seguintes:
 - Custos de Aquisição:
 - Custos de consultoria.
 - Custos de licença e compra de equipamentos.
 - Custo de instalação de equipamentos.
 - Custo para modificar o local do equipamento (ar-condicionado, segurança, etc.)
 - Custos de Iniciação:
 - Custos para iniciar a utilização do software.
 - Digitalizar registros antes em meio impresso.
 - Treinamento de pessoal para operar o software.
 - Custo de instalação de aparatos de comunicação (roteadores, linhas de dados, etc).
 - Custo de pessoal responsável pelo funcionamento do sistema.

8

2. Análise de Sistemas

2.4. Análise Econômica (2)

- Custos Relacionados ao Projeto:
 - Custo de aplicativos comprados.
 - Custo de modificações de software para adaptações a sistemas locais.
 - Custo de treinar o pessoal que oferece suporte ao usuário na utilização do aplicativo.
- Custos Permanentes:
 - Custos de manutenção do sistema (hardware, software e facilidades).
 - Custos de aluguel de equipamentos e estruturas (linhas de dados, hardware, etc.).
 - Custos da depreciação do hardware.

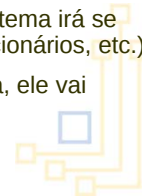


9

2. Análise de Sistemas

2.5. Diagrama de Arquiteturas (1)

- A última etapa da “Análise de Sistemas” consiste em tentar imaginar como deverá ser a estrutura do sistema.
- Essa tentativa de “estruturar” o programa é chamada de modelagem.
- Para modelar a arquitetura de um sistema, normalmente considera-se cinco regiões de processamento, ou seja:
 - 1 – Interface com o usuário
 - 2 – Processamento de entrada
 - 3 – Funções de processo e de controle
 - 4 – Processamento de saída
 - 5 – Manutenção e auto teste
- Para fazer a modelagem de um sistema, podem ser utilizados Diagramas de Contexto da Arquitetura, ou simplesmente DCAs.
- Na prática, um DCA é uma figura que mostra como o sistema irá se relacionar com o ambiente (a fábrica, a indústria, os funcionários, etc.).
- Um DCA segue sempre o mesmo padrão básico, ou seja, ele vai mostrar o relacionamento entre o programa básico e:
 - O usuário
 - Dados de entrada
 - Dados de saída
 - Manutenção e autoteste (se necessário)

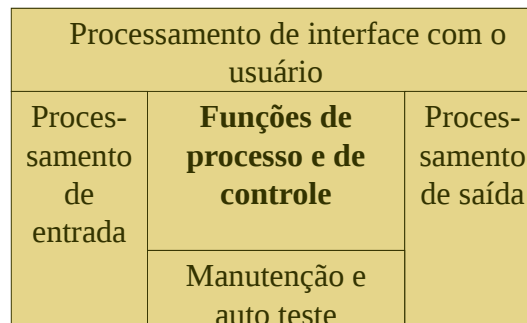


10

2. Análise de Sistemas

2.5. Diagrama de Arquiteturas (2)

- A figura abaixo é um modelo que demonstra a estrutura básica de um DCA.



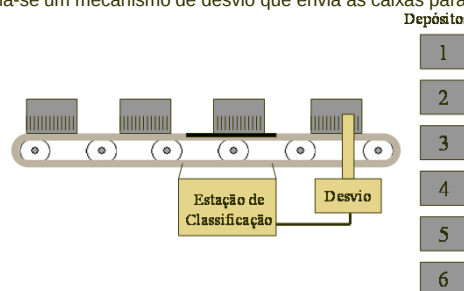
- Após modelar o software utilizando um DCA, esse modelo deve ser mais detalhado.

11

2. Análise de Sistemas

2.5. Diagrama de Arquiteturas (3)

- Exemplo
 - Software que realiza a classificação de caixas que se movimentam ao longo de uma correia transportadora em uma indústria. O cenário é o seguinte:
 - Cada caixa contém um código de barras que identifica o conteúdo.
 - Conforme o código, ela deve ir para um entre 6 depósitos diferentes.
 - As caixas passam por uma estação de classificação que contém um leitor de código de barras e um computador.
 - O software receberá dados do leitor de código de barras. Conforme código recebido, aciona-se um mecanismo de desvio que envia as caixas para o depósito correto.

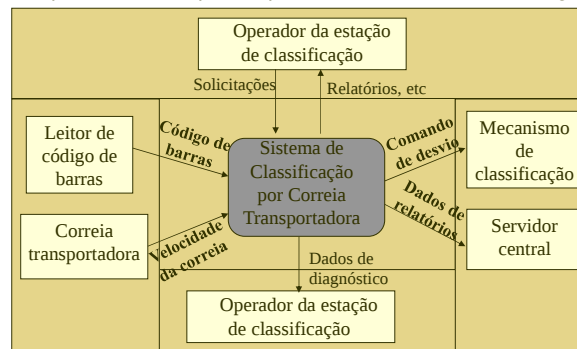


12

2. Análise de Sistemas

2.6. Diagrama de Arquiteturas (4)

- Considerando esse exemplo, o diagrama DCA vai mostrar qual é o contexto, ou cenário, em que o software está inserido. Ou seja, a interface entre o software e as entidades externas a ele.
- Vai mostrar, portanto, sua interface com a correia transportadora, o leitor de código de barras, o operador do software (o funcionário que vai utilizar o software), o mecanismo de desvio e o computador da máquina.
- Portanto, para este exemplo um possível DCA é mostrado a seguir:



13

3. Análise de Requisitos

3.1. Introdução

- Como visto anteriormente, a fase de análise de software é dividida em duas, ou seja:
 - **Análise de Sistema.**
 - **Análise de Requisitos.**
- **Análise de Sistemas:**
 - Se concentra em todos os elementos do sistema, não apenas no software.
 - **Objetivos:** identificar necessidades dos usuários, análise econômica, análise técnica, restrições de prazo e custo, etc.
- **Análise de Requisitos:**
 - Se concentra **apenas** no software.
 - Objetivos:
 - Rever elementos problemáticos utilizando a especificação gerada durante a fase de análise de sistemas.
 - Refinar os dados de entrada/saída.
 - Aprimorar modelos, DCAs.

14

3. Análise de Requisitos

3.1. Introdução



A análise de requisitos age como uma ponte entre a "Análise de Sistema" e a fase de "Projeto de Software".

- A análise de requisitos de software pode ser dividida em cinco áreas de esforço, ou seja:

1. Reconhecimento do Problema
2. Avaliação e Síntese
3. Modelagem
4. Especificação
5. Revisão



15

3. Análise de Requisitos

3.2. Aspectos da Análise de Requisitos (1)

1. Reconhecimento do Problema

- Entender qual é o problema que será resolvido pelo software.

2. Avaliação e Síntese

- Análise de todas as funções a serem desempenhadas pelo software. Não mais de forma geral, mas clara e específica.
- Identificar restrições do projeto, tais como: o programa deve rodar em Windows, deve rodar em estações SUN, deve buscar um registro em menos de 1 ms, etc.

3. Modelagem

- Durante a fase de avaliação e síntese, vão sendo então produzidos (ou aprimorados) modelos, como diagramas ou protótipos.

4. Especificação

- Deve ser produzido um documento chamado "Especificação dos Requisitos", como um esforço conjunto entre o desenvolvedor e o cliente.

5. Revisão

- A partir do documento produzido, faz-se então uma revisão, verificando-se se o cliente acha necessário modificações.
- Logo que a Revisão for concluída, a "Especificação dos Requisitos" de Software é "assinada" pelo cliente e pelo desenvolvedor.
- A especificação torna-se um "contrato" de desenvolvimento de software.
- Mudanças solicitadas depois que a Especificação for concluída serão consideradas, porém, cada mudança posterior pode aumentar o custo e/ou alongar o prazo de entrega.

16

3. Análise de Requisitos

3.2. Aspectos da Análise de Requisitos (2)

- **Requisitos Funcionais:**

- São declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como se deve comportar em determinadas situações.

- **Requisitos Não Funcionais:**

- São aqueles que não dizem respeito diretamente às funções específicas fornecidas pelo sistema.
- Eles podem estar relacionados às propriedades do sistema, tais como confiabilidade, tempo de resposta e espaço em disco.

17



3. Análise de Requisitos

3.3. Documento de Espec. de Requisitos

- Norma elaborada pelo IEEE (Standard 830/98):

Índice

1. Introdução

1.1 Propósito

(Deve delinear o propósito do documento)

1.2 Escopo do Produto

(Deve identificar pelo nome o software a ser produzido; deve explicar o que o software fará e, se necessário, o que ele não fará)

1.3 Definições, Siglas e Abreviações

(Deve fornecer as definições de todos os termos, acrônimos e abreviações usados no documento, para que se possa interpretá-lo adequadamente)

1.4 Referências

(Deve fornecer as referências a outros documentos que foram utilizados para embasamento)

1.5 Visão Geral da ERS

(Deve descrever resumidamente o que terá no restante do documento e como o mesmo está organizado)

18



3. Análise de Requisitos

3.3. Documento de Espec. de Requisitos

2. Descrições Gerais

(Descreve os fatores gerais que afetam o produto e seus requisitos. Esta seção não declara requisitos específicos)

2.1 Perspectivas do Produto

(Deve colocar o produto em perspectiva com outros produtos relacionados. Se o produto é independente e totalmente autocontido, esse fato deve ser declarado aqui)

2.2 Funções do Produto

(Deve fornecer um resumo das maiores funcionalidades que o software deve realizar)

2.3 Características do Usuário

(Deve descrever as características gerais dos usuários do produto, incluindo nível de educação, experiência e conhecimento técnico)

2.4 Restrições Gerais

(Deve descrever as restrições que o sistema tem)

2.5 Suposições e Dependências

(Deve listar cada um dos fatores que afetam os requisitos declarados na Seção 3)

3. Requisitos Específicos

Apêndices

19

3. Análise de Requisitos

3.3. Documento de Espec. de Requisitos

3. Requisitos Específicos

3.1 Requisitos Funcionais

3.1.1 Requisito Funcional I

3.1.1.1 Introdução

3.1.1.2 Entradas

3.1.1.3 Processamento

3.1.1.4 Saídas

3.1..n Requisito Funcional n

3.2 Requisitos da Interface Externa

3.2.1 Interface com Usuário

3.2.2 Interface Hardware

3.2.3 Interface Software

3.1.4 Interface Comunicação

3.3 Requisitos de Desempenho

3.4 Limitações de Projeto

3.4.1 Padrões

3.1.2 Limitações de Hardware

3.5 Atributos de Qualidade

3.5.1 Disponibilidade

3.5.2 Segurança

3.5.3 Manutenibilidade

3.5.n

3.6 Outros Requisitos

3.6.1 Base de Dados

3.5.2 Operações

3.5.3 Adaptação ao Local

3.5.n

Link Para Acessar a Norma Completa:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720574>

Observação: deve ser acessada a partir da rede da USP.

20