RESUMO / REVISÃO DE ESW – ENGENHARIA DE SOFTWARE

LEGENDA:

Conceitos;

Vantagens / Desvantagens;

Exemplos;

Comentários;

Importância (Certeza que vai cair na prova);

AULA 09 – TERMO DE ABERTURA, DECLARAÇÃO DE ESCOPO E DOCUMENTO DE VISÃO

Requisito: consiste na definição documentada de uma propriedade ou comportamento que um produto ou serviço particular deve atender.

Restrições: são fatores internos e externos associados ao escopo do projeto que limitam as opções de equipe de gerenciamento do projeto (em geral são chamados de requisitos obrigatórios). Podem ser técnicos ou não.

Exemplo de Restrições de Hardware:

- Projeto requer uma determinada tecnologia, tal como WebServices.

Exemplo de Restrições de Software:

- Todos os softwares deverão ser desenvolvidos utilizando a linguagem C#.

Exemplo de Restrições de Ambiente e Tecnologia:

- O projeto conta com a participação de três integrantes que dividirão as atividades de desenvolvimento.

Premissas: são suposiçoes externas ao projeto adotadas como verdadeiras para efeito de planejamento do projeto. É algo que normalmente não está sob controle interno do projeto.

Exemplo de Premissa:

- Durante o período de execução, o recurso humano que desenvolve o software de gerência não será deslocado para outro projeto.

Risco: significa algo que pode dar errado. A ocorrência dele pode comprometer o andamento do projeto.

Exemplo de Risco:

- Requisitos não compreendidos;
- Mal desenho do projeto;
- Erro na implementação...

Riscos de Projeto: cronograma, pessoal e orçamento;

Riscos Técnicos: análise, design, implementação e testes. Ferramentas de Hardware e Software;

Riscos de Negócios: mudanças no mercado, novas estratégias. Requisitos e restrições organizacionais.

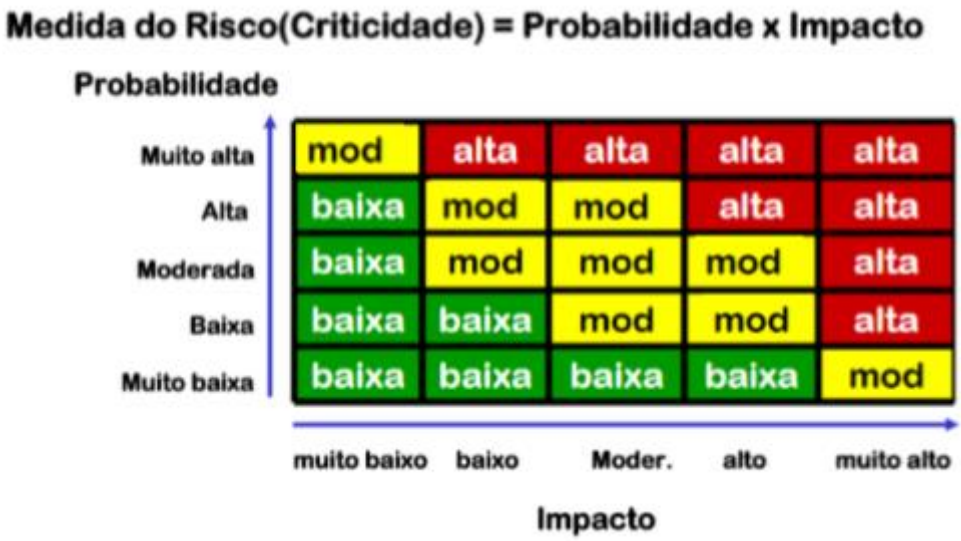
Análise de Risco: é o processo de identificar, analisar e responder a estes eventos.

Identificação dos Riscos: é o levantamento dos eventuais riscos de diferentes naturezas.

Estimativas: baseia-se no julgamento, na intuição e na experiência em estimar probabilidades de ocorrência de potenciais riscos e medir a intensidade de perdas e ganhos potenciais (impacto sobre o projeto).

Prioridade: (Probabilidade x Impacto)

Valor Esperado: permite definir uma priorização dos riscos do projeto.

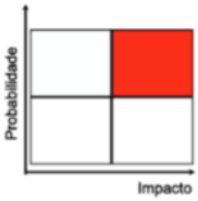


Escala ordinal – muito baixa, baixa, moderada, alta, muito alta

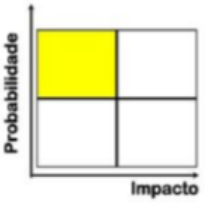
Escala Cardinal – assinala valores numéricos. Pode ser valores lineares (.1/ .3/ .5/ .7/ .9)

Referencial	Probabilidade de Ocorrência
Muito alta	0.90
Alta	0.70
Moderada	0.50
Baixa	0.30
Muito baixa	0.10

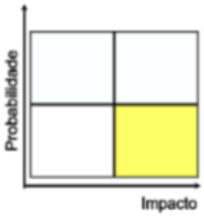
1º Quadrante: nenhum projeto sobrevive com riscos em nível crítico a longo prazo. (Ações de mitigação são necessárias para a saúde do projeto).



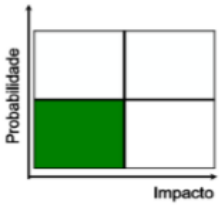
2º Quadrante: estes riscos frequentemente são relacionados às operações cotidianas. Se não forem gerenciados a tempo podem ser tão impactantes quanto os do 1º Quadrante. São chamados de "comedores de recursos". (Devem ser monitorados constantemente).



3º Quadrante: estes riscos são frequentemente guiados por fatores externos ou ambientais, fora do controle da gerência tais como terremotos ou furacões. (A contigência é mais apropriada para estes riscos).



4º Quadrante: geralmente são aceitáveis em seu nível atual. (Devem ser monitorados, porém, com menor frequência).



Gerenciamento de Risco: reduzir a probabilidade de ocorrência e o impacto potencial de cada risco. (- ações para redução dos riscos | - ações, em caso da ocorrência, para minimizar o impacto).

Evitar: significa eliminar o risco pela escolha de uma ação diferente;

Mitigar: envolve tomar medidas para reduzir a probabilidade de que o evento de risco ocorra ou reduzir o impacto potencial;

Aceitar: significa lidar com ele, se e quando ocorrer, em vez de tomar medidas para evitar ou reduzir o impacto;

Revisões: monitoramento e reavaliação de todos os riscos e determinar, se houver, novos riscos.

Ferramentas de Gerenciamento de Risco:

FMEA (Failure Mode and Effective Analysis): é uma ferramenta versátil que pode ser usada para avaliar problemas na produção de produtos ou em processos, sendo chamada, nestes casos de PFMEA (Process FMEA) | Possui 3 indicadores numéricos:

- Severidade;

- Ocorrência;
- Detecção.

PMBOK (Project Management Body of Knowledge): descreve ferramentas, técnicas e as melhores práticas a serem adotadas no gerenciamento de projetos. A análise de risco é feita das formas:

- **Qualitativa:** priorização dos riscos com maior probabilidade de impactar nos resultados do projeto.
- **Quantitativa:** efetua a análise numérica do efeito dos riscos identificados nos objetivos gerais do projeto.

Diagrama de Ishikawa (Espinha de Peixe): leva em conta os 6 M's e o efeito de cada um deles (Método, Mão de Obra, Material, Medida, Meio Ambiente e Máquina).

Termo de Abertura do Projeto: é o documento que autoriza formalmente o início de um projeto na organização.

- Formaliza o início do projeto;
- Dá autoridade necessária ao gerente de projetos.

Contém:

- Principais responsáveis, requisitos iniciais, principais entregas, premissas e restrições.
- Possui informações de alto nível e declarações em linguagem natural (ou seja, que clientes podem ler e entender).

Documento de Visão: tem como objetivo fornecer uma visão geral do sistema, abrangendo coleta, análise e definição das necessidades e recursos necessários aos envolvidos e usuários finais.

- Estabelece subsídios para a modelagem do sistema;
- Seu propósito é expor as necessidades e funcionalidades gerais do sistema;
- Possui detalhes de como o sistema satisfará essas necessidades

- Será desenvolvido após a coleta e análise dos requisitos preliminares feitos no:
    - Documento de Requisitos;
    - Documento de Regras de Negócio.
- Seu foco está nas necessidades dos stakeholders (as partes interessadas);

Sua estrutura é separada em:

1 – Escopo | 2 – Gestores | 3 – Levantamento de Necessidades | 4 – Classificaçao de Necessidades por Categoria | 5 – Funcionalidade do Produto | 6 – Integração com Outros Sistemas | 7 – Restrições.

Declaração de Escopo: delimita a abrangência do projeto a partir de análises de “como, quando, onde e para quem”.

- Consiste na geração e entrega de produtos e serviçoes;
- Informações devem ser detalhadas e progressivamente elaboradas.

É bem definida e identifica os stakeholders e define limites claros quanto ao que estará dentro e fora do projeto.

Escopo do Produto: são características e funcionalidades que caracterizam o produto ou serviço. (medido através dos requisitos do produto).

Escopo do Projeto: é todo o trabalho que terá que ser realizado para produzir o produto ou serviço. (medido através do plano de gerenciamento do projeto).

Sua estrutura é separada em:

1 – Objetivos do Projeto | 2 – Objetivos do Negócio | 3 – Descrição do Projeto | 4 – Marcos do Projeto | 5 – Riscos Iniciais | 6 – Orçamento Inicial | 7 – Aprovações.

LEGENDA:

Conceitos;

Vantagens / Desvantagens;

Exemplos;

Comentários;

Importância (Certeza que vai cair na prova);

AULA 10 – MAPEAMENTO DE PROCESSOS

Processo: é um grupo de atividades realizadas numa sequência lógica com o objetivo de produzir um bem ou serviço que tem valor para um grupo específico de clientes (Hammer e Champy, 1994). (ENTRADA -> PROCESSO -> SAÍDA).

- Ajudam a implementar a estratégia nas operações do negócio;
- São ativos de grande valor para a organização;
- Refletem como a empresa funciona;
- São responsáveis pela criação de valor na perspectiva do cliente.
- **São a ponte entre T.I. e Negócio.**

Hierarquia:

Macroprocesso: envolve mais de uma função na estrutura organizacional e sua operação tem impacto significativo no modo como a organização funciona.

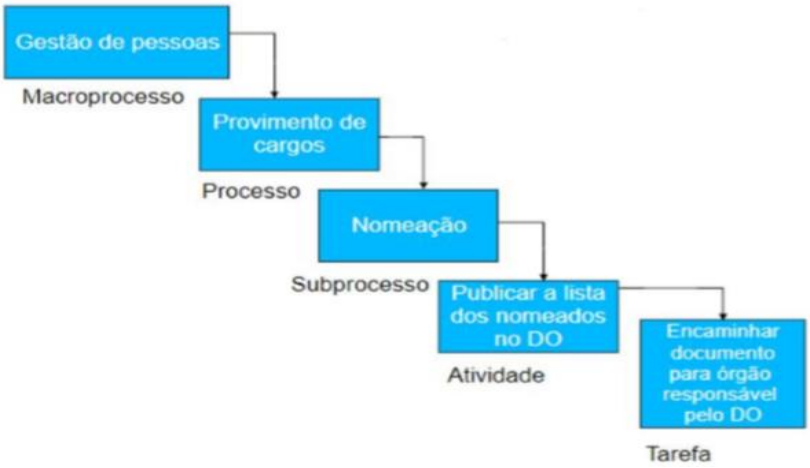
Processo: é criado dentro de uma estrutura hierárquica (definição já estabelecida no início).

Subprocesso: é um processo que está incluso em outro (caso o processo deixe de existir os subprocessos deixarão de existir também).

Atividade: são ações a serem realizadas dentro de um processo ou subprocesso para produzir um resultado particular.

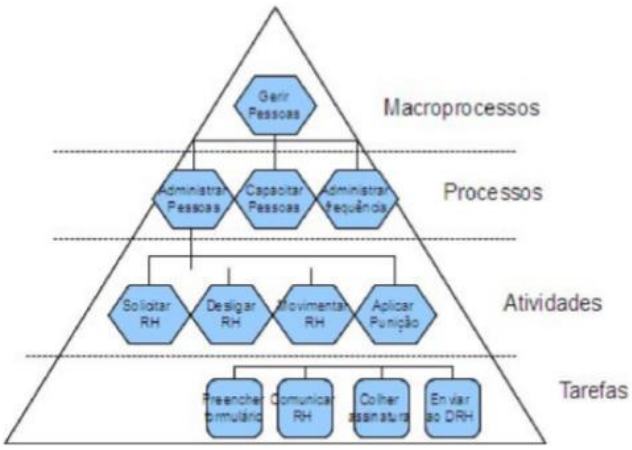
Tarefa: são elementos individuais e/ou subconjuntos de uma atividade.

Exemplo de uma Hierárquia de Processo de Gestão de Pessoas (DO – Diário Oficial).



Visão de Processo:

I. Visão Top Down: visão partindo do macro processo até as tarefas.



- Abordagens de cima para baixo enfatizam o planejamento e uma compreensão completa do sistema;
- **Exemplo:** quando o cliente diz o que deseja para o projeto e dai informamos o valor estamos aplicando o método Top Down.
- Nenhum código pode começar até que um nível de detalhe necessário tenha sido atingido na concepção do sistema.

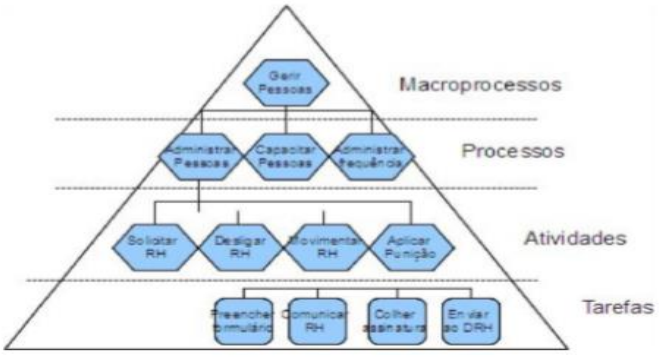
Vantagens:

- Normalmente é usada na fase inicial do projeto;
- Mais rápido para ser informado ao cliente;
- Útil para medir o interesse do cliente no projeto;
- Utilizada em projeto quando não temos informações detalhadas.

Desvantagens:

- Baseado em experiências anteriores;
- Deve ser realizado por um especialista;
- Possui grande risco ao estimar o custo do projeto.

II. Visão Bottom Up: visão partindo da tarefa para o macroprocesso.



- Abordagens de baixo para cima dá foco à codificação e testes iniciais, o que pode começar assim que o primeiro módulo foi especificado;
- Corre o risco de que os módulos sejam codificados sem ter uma ideia clara da ligação entre eles;
- Identifica as prioridades das atividades.

Vantagens:

- Maior precisão;
- Desenvolve o comprometimento da equipe;
- Baseado na análise detalhada do projeto;
- Fornece a base para monitoramento e controle do projeto.

Desvantagens:

- Maior esforço de tempo, custo e recursos para desenvolver a estimativa;
- Tendência do time “inflar” as estimativas;
- Requer que o projeto esteja bem definido e entendido.

O método Top Down é vantajoso para uma primeira avaliação do projeto e também para captar as expectativas do cliente, e o Bottom Up é extremamente recomendado antes do início da execução do projeto, podendo validar as informações mais precisas com o cliente e consequentemente diminuindo os riscos relacionados a parte financeira do projeto.

Mapeamento de Processos: é uma técnica geral que representa cada passo de operação de uma unidade empresarial em termos de entradas, saídas e ações.

- Seu objetivo é fazer a empresa enxergar os pontos fortes e fracos, melhorar o entendimento sobre os processos e aumentar a performance do negócio;
- Alguns passos para modelar o mapeamento de processos podem ser feitos através de ferramentas administrativas como Diagrama de Ishikawa e 5W2H

Tipos de Modelo de Mapeamento de Processo:

Modelo AS IS: gerado na etapa Identificação e Mapeamento de Processos. Seu objetivo é obter uma formalização sobre o fluxo do processo de negócio e como é realizado na situação atual em que é executado na organização.

- Não devem participar chefias em geral (apenas as pessoas do dia a dia do trabalho);
- Tempo de vida curto (não investir muito na sua documentação);
- Pode ser realizado através de: entrevistas, questionários, observação e reunião.

Modelo TO BE: gerado na etapa Redesenho de Processos. É uma evolução do Modelo AS IS no qual são reavaliadas as questões de negócio envolvidas

buscando-se, através de melhorias culturais e organizacionais, maior eficiência na execução do processo.

- Devem participar chefias em geral;
- Pode ser realizado através de: entrevistas e reunião.

BPMN (Business Process Modeling Notation): é uma notação da metodologia de gerenciamento de processos que fornece uma série de ícones padrões para o desenho deles, facilitando o entendimento do usuário.

- Seu objetivo é dar suporte ao gerenciamento de processos de negócio;
- É um padrão internacional de modelador de prcoessos aceito pela comunidade;
- É independente de qualquer metodologia de modelagem de processos.

Pool: representa um processo ou uma entidade;

Lane(Raia): é uma subparte da pool, são usadas para organizar e categorizar;

Fluxo de Sequência: usado para mostrar a ordem que as atividades serão executadas;

Fluxo de Mensagem: usado para mostrar o fluxo de mensagem entre duas Pools;

Associação: usada para associar atividades com objetos;

Atividades: tarefa simples que é usada quando não pode ser dividida em mais detalhes;

Subprocesso: atividade composta onde os detalhes são definidos em um novo fluxo;

Evento de Início: utilizado ao início do processo (cada processo tem apenas um início);

Evento Intermediário: acontece durante o fluxo do processo (pode ter vários intermediários);

Evento Fim: finaliza o fluxo (um processo pode ter mais de um evento fim);

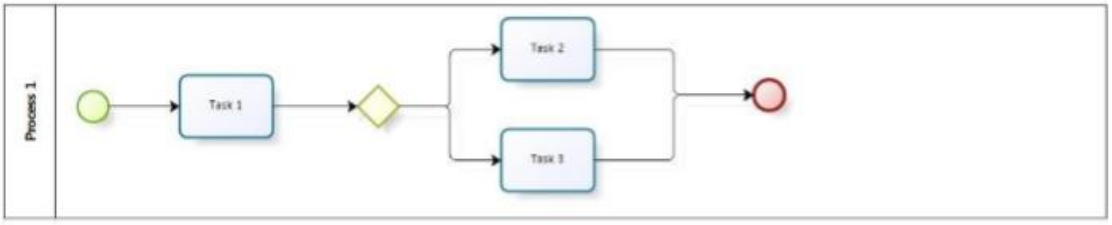
Gateway (decisão): usado para controlar as ramificações e fluxos do processo;

Anotação: usada para fornecer informações adicionais para facilitar a leitura do diagrama por parte do usuário;

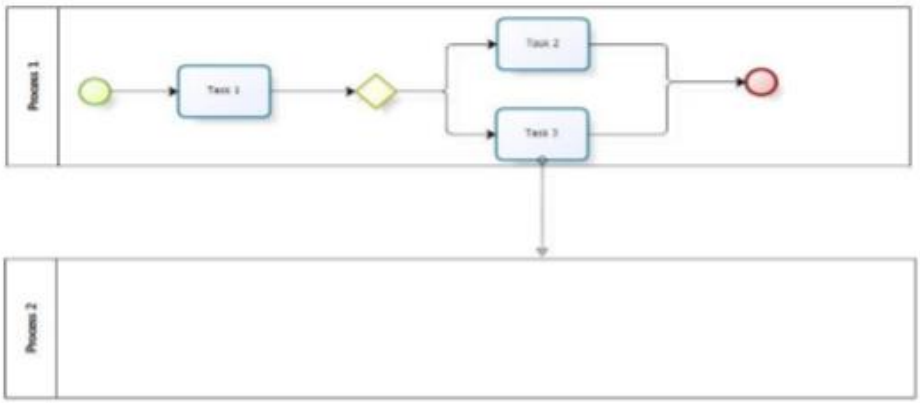
Objetos de dados: fornece informações sobre o que a atividade precisa para ser executada.

Tipos de Diagramas de Processo de Negócio:

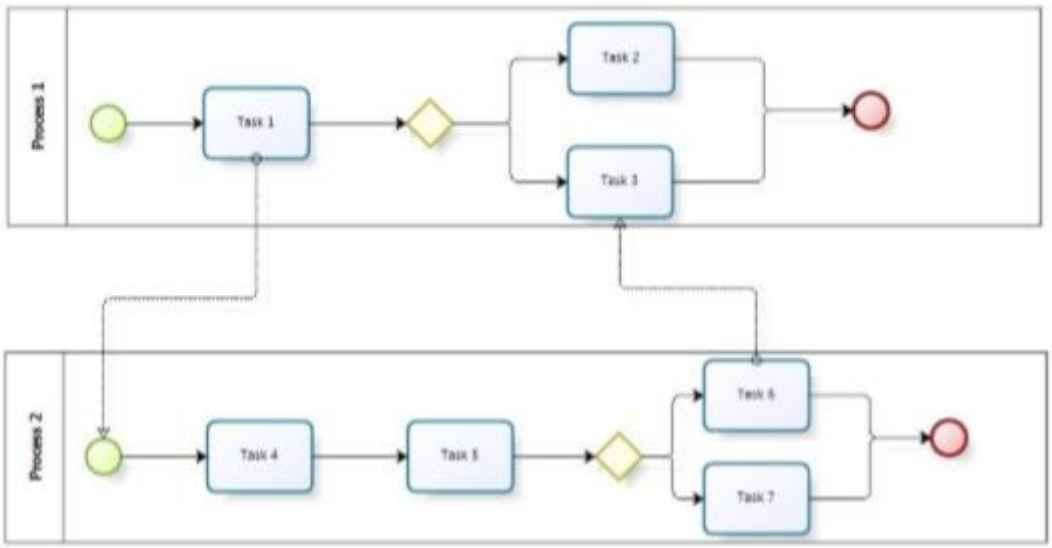
Privativo ou Privado: quando não interessa a interação desse processo com outros com os quais ele possa interagir. Preocupa-se com a forma deste fluxo em si.



Abstrato: representa a interação entre um processo principal e outro processo participante. Em relação ao participante, não há preocupação com o conteúdo do fluxo em si, **mas sim como ele colabora com os outros fluxos dentro de um sistema.**



Colaborativo: descreve a **interação entre duas ou mais entidades do negócio,** sendo que o contéudo do fluxo é especificado em todas as entidades.



LEGENDA:

Conceitos;

Vantagens / Desvantagens;

Exemplos;

Comentários;

Importância (Certeza que vai cair na prova);

AULA 11 – EAP – ESTRUTURA ANALÍTICA DE PROJETO

Gestão de Projetos: é um conjunto de práticas que serve de guia a um grupo para trabalhar de maneira produtivas. **Em ESW:** compreende atividades que visam assegurar que o sistema ou produto de software seja entregue ao cliente no prazo pré-definido e esteja de acordo com os requisitos definidos por ele.

- Possui três pilares: I – foco no cliente , II – fazer a equipe trabalhar de forma produtiva e colaborativa, III – administrar os recursos do projeto

EAP e Cronograma: são ferramentas que vão ajudar a execução de um projeto de forma eficaz e organizada.

- EAP é feita antes do cronograma.

EAP(Estrutura Analítica de Projeto) ou WBS (Work Breakdown Structure): é uma ferramenta visual que permite a estruturação de um projeto de forma simples e contém todo o trabalho necessário para conclusão dele.

- O que não está no EAP não faz parte do escopo do projeto;
- Deve cobrir 100% do escopo do projeto e capturar todos os entregáveis relevantes.

Vantagens:

- É possível alocar um orçamento para níveis superiores e também pensarn os recursos necessários;
- Pode identificar riscos, que podem ser avaliados e monitorados durante a execução do projeto;
- Facilita a comunicação entre os stakeholders (partes interessadas);
- Facilita a definição e o controle do que deve ser executado;
- Ferramenta utilizada no planejamento do projeto;
- Viabiliza a elaboração do cronograma do projeto.

Para fazer o EAP é preciso que:

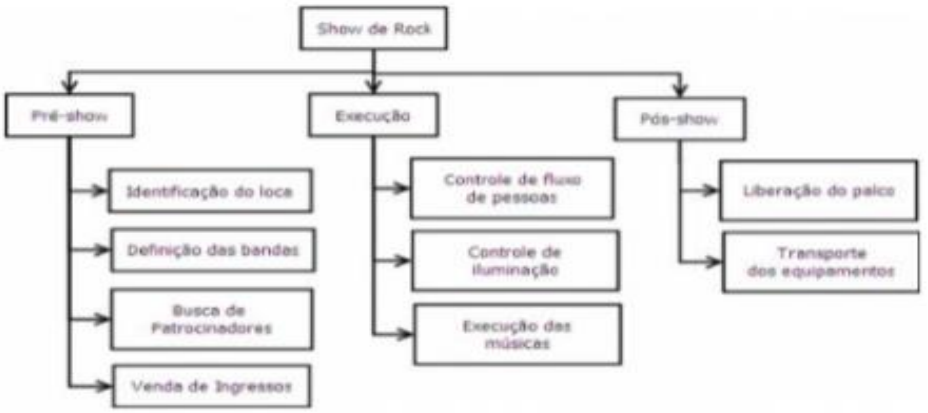
- Todas as informações relevantes sejam consideradas;
- Tenha os pacotes de trabalho;
- A EAP não deve conter atividades, logo não pode ser representada por verbos mas sim substantivos;

- Pode ser representada de forma gráfica ou descritiva.

Características:

- Organiza e define o escopo total e representa o trabalho especificado na atual declaração de escopo do projeto aprovado;
- Deve ser completa, organizada e pequena o suficiente para tornar possível a medição do progresso, mas não detalhada o suficiente para se tornar um obstaculo;

EAP por Fases: organiza fases no primeiro nível e eventualmente no segundo nível também.



Vantagens:

- Oferece uma visão “cronológica” dos acontecimentos do projeto;
- Facilita o entendimento de pessoas leigas;
- Facilita o posterior gerenciamento das atividades.

Desvantagens:

- Pode ofuscar a visão das partes necessárias para uma entrega específica;
- Tende a incentivar que se incluam atividades administrativas (ex: controle do projeto).

EAP por Entregas: mostra as partes necessárias para compor as entregas do projeto.

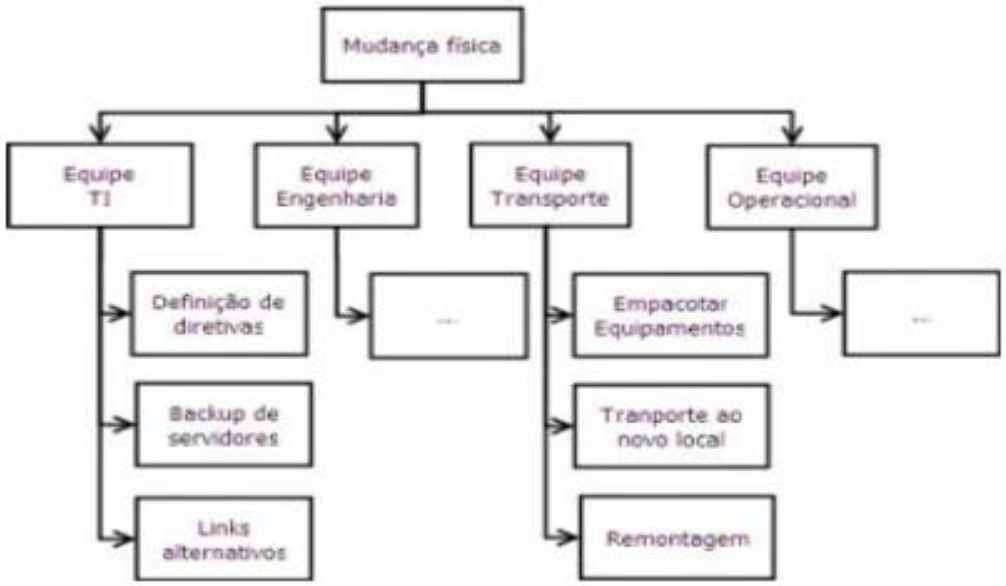
Vantagens:

- Visualiza claramente as partes que compõe o projeto;
- Facilita a discussão técnica e os caminhos alternativos;
- Failita a identificação de riscos técnicos.

Desvantagens:

- Não oference visão cronológica.

EAP por Equipes: visualiza os pacotes de trabalho a partir da divisão de equipes do projeto.



Vantagens:

- Ótima para ocasiões em que o projeto tem equipes com responsabilidades muito diferentes.

Desvantagens:

- Não mostra cronologia nem a organização das partes das entregas.

LEGENDA:

Conceitos;

Vantagens / Desvantagens;

Exemplos;

Comentários;

Importância (Certeza que vai cair na prova);

AULA 12 – CRONOGRAMA

Cronograma: é uma representação gráfica do tempo investido em uma determinada tarefa ou projeto, segundo as tarefas que devem ser executadas no âmbito desse projeto. Seu objetivo é medir o desempenho das equipes multidisciplinas envolvidas no projeto, e ainda desenvolver meios eficazes para melhorar a sua evolução.

- Chamado também de “Mapa do Tempo”;
- Antes de definir as tarefas deve-se pensar: o que, como e quando fazer as atividades.

Método do Caminho Crítico (Critical Path Method – CPM): basicamente se refere ao caminho mais “longo” do projeto, nele, calcula-se as datas teóricas de **início e término mais cedo e mais tarde** de todas as atividades do cronograma, sem considerar quaisquer limitações de recursos, realizando uma análise do **caminho de ida e de volta.**

- É o **caminho mais longo,** possui **folga total nula** e, portanto, determina o **maior tempo para conclusão do projeto.**

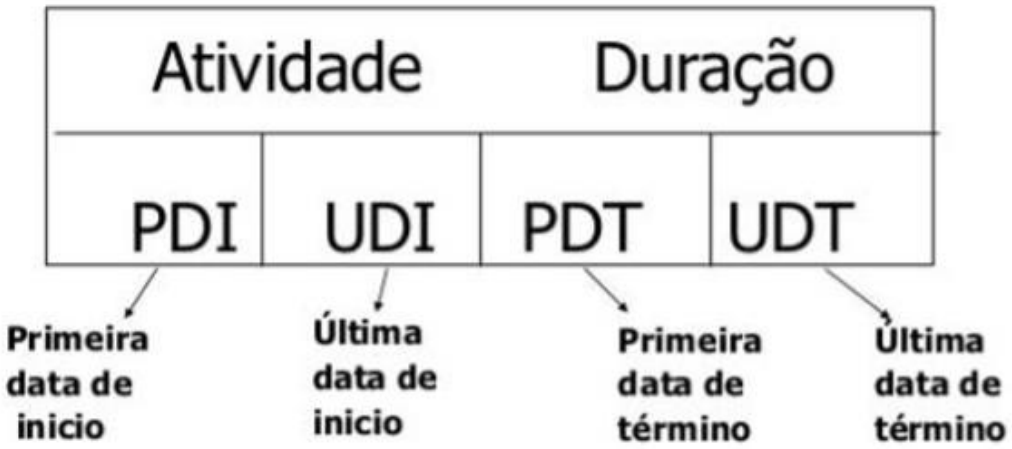
Atividades Críticas: são as atividades do cronograma em um caminho crítico.

PDI: Primeira data de Início;

PDT: Primeira Data de Término – PDT = PDI + duração da atividade;

UDI: Última data de Término;

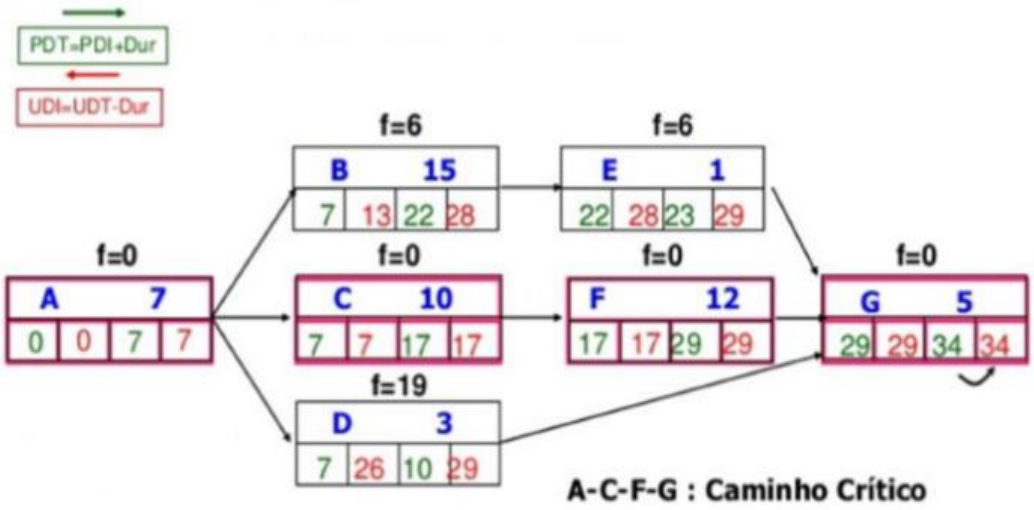
UDT: Última Data de Início – UDI = UDT – duração da atividade.



Folga Total: é o atraso total permitido para a data de início de uma atividade do cronograma sem atrasar a data de término do projeto ou violar alguma restrição dele.

Exemplo:

Atividade	Sucessora	Duração
A	B,C,D	7
B	E	15
C	F	10
D	G	3
E	G	1
F	G	12
G	---	5



- Tempo de duração e quantidade de atividades pode ser um fator determinante para o caminho crítico.
- Duração Total (nesse exemplo): 34 unidades de tempo (dias)
- No caso de uma atividade possuir mais de uma predecessora adota-se o maior valor (PDI = máx PDT);
- No caso ela tenha mais de uma sucessora adota-se o menor das UDI das sucessoras (UDT = min UDI);

LEGENDA:

Conceitos;

Vantagens / Desvantagens;

Exemplos;

Comentários;

Importância (Certeza que vai cair na prova);

AULA 13 – UML: CASOS DE USO

UML: é uma linguagem de modelagem que permite representar um sistema de forma padronizada.

Vantagens:

- Permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados;
- Qualquer sistema, seja qual for o tipo, modelado corretamente, será:
  - Consistente;
  - Fácil de se comunicar com outras aplicações;
  - Simples de ser atualizado;
  - Compreensível.

Análise de Requisitos: o diagrama de casos de uso mostrará o que os atores externos do futuro sistema deverão esperar dele, conhecendo toda sua funcionalidade sem se importar como está será implementada.

Análise: só serão modeladas classes que pertençam ao domínio principal do problema do software.

Design (Projeto): resulta no detalhamento das especificações para a fase de programação do sistema.

Codificação (Programação): a programação é uma fase separada e distinta onde os modelos criados são convertidos em código.

Teste: Unidade: são para classes individuais | Integração: são aplicados já usando classes e componentes integrados para se confirmar se elas estão cooperando umas com as outras | Aceitação: observam o sistema como uma "caixa preta" e verificam se está funcionando como o especificado nos primeiros diagramas de caso de uso.

Implantação: são utilizados para representar a arquitetura física de um sistema. Há dois tipos: diagramas de componentes e de implantação.

Caso de uso:

- Não descreve como o software deverá ser construído, mas sim como ele deverá se comportar;
- Pode "incluir" **(DEVE)** outra funcionalidade ou "estender" **(PODE)** outro caso de uso com seu próprio comportamento.
- Um ator é um humano, entidade ou dispositivos que interagem com o sistema para executar um trabalho

Sequência de eventos:

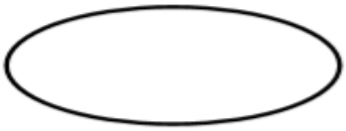
- Ação do ator: ações numeradas;
- Resposta do sistema: descrição numerada das respostas;
- Sequências alternativas: descrição de exceções por número de linha(s).

Elementos do Diagrama de Casos de Uso:

Ator: representa o papel de uma entidade externa ao sistema. Os atores iniciam a comunicação com o sistema através dos casos de uso.

Caso de Uso: representa uma sequência de ações executadas pelo sistema e um conjunto de serviços que ele fornece aos usuários.

- **Caso de Uso**



Função

Relacionamento: união de um ou mais elementos através de associações, tanto atores quanto casos de uso podem possuir relacionamentos.

Relacionamento

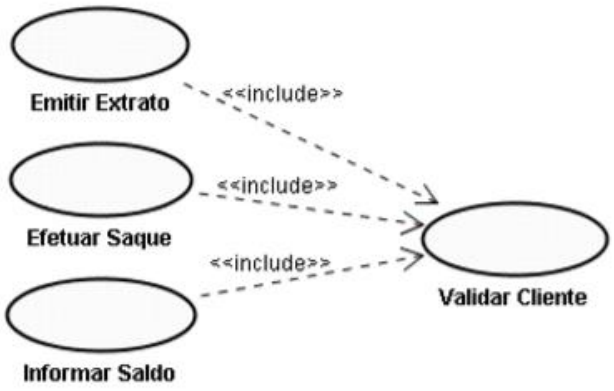
Sistema: projeto ou módulo do projeto.

Associação: entre um ator e um caso de uso significa que estimulos podem ser enviados entre eles.

Inclusão:

- Utilizado quando um caso de uso é usado dentro de outro;
- Os relacionamentos de inclusão indicam **OBRIGATORIEDADE;**
- A execução do primeiro obriga a do segundo.

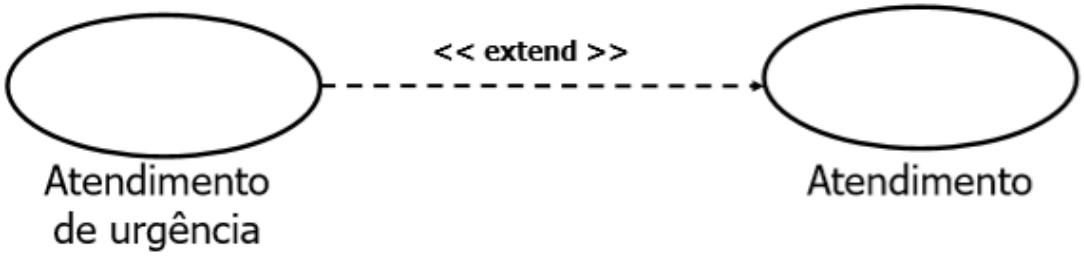
Exemplo:



Extensão:

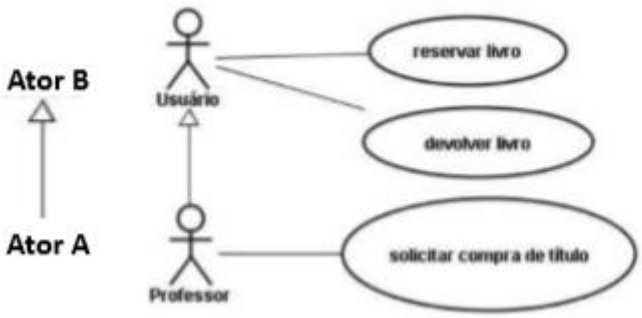
- Utilizado quando se tem dois casos de uso que fazem algo parecido, só que o B faz alguma coisa a mais que A. B estende A;
- B representa alguma situação não muito comum que ocorre em A mediante a satisfação de uma pré-condição;
- Caso o A seja executado o B **PODERÁ** ser executado também.

Exemplo:



Generalização / Especialização (Generalization):

- Um relacionamento entre um caso de uso genérico para um mais específico, que herda todas as características de seu pai;
- É do generalizador para o generalizado;
- Uma generalização de um ator A para B significa que o ator A pode se comunicar com os mesmos casos de uso que o ator B.



LEGENDA:

Conceitos;

Vantagens / Desvantagens;

Exemplos;

Comentários;

Importância (Certeza que vai cair na prova);

AULA 14 – UML

Diagramas Estáticos – DE | Diagramas Dinâmicos - DD

UML: é uma linguagem de modelagem que permite representar um sistema de forma padronizada;

Vantagem:

- Permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados.

DE Casos de Uso:

- É o mais geral;
- Devido a sua linguagem simples, os usuários podem ter uma ideia geral de como o sistema irá se comportar;
- O objetivo é auxiliar a comunicação entre os analistas e o cliente;
- Descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário;
- Notação: atores / casos de uso / relacionamentos: associações, generalizações / extends e includes.

DE Classes:

- É um dos mais utilizados e é um dos mais importantes da UML. Serve de apoio para a maioria dos diagramas;
- Define a estrutura das classes utilizadas pelo sistema;
- O Objetivo é descrever os tipos de objetos no sistema e o relacionamento entre eles;
- Contém: Classes / Relacionamento / Notações

Classes: graficamente são representadas por retângulos contendo atributos e métodos.

Atributos: representam o conjunto de características (estado) dos objetos daquela classe

- Visibilidade: + público | # protegido | - privado

Métodos: representam o conjunto de operações (comportamento que a classe fornece)

- Visibilidade: + público | # protegido | - privado

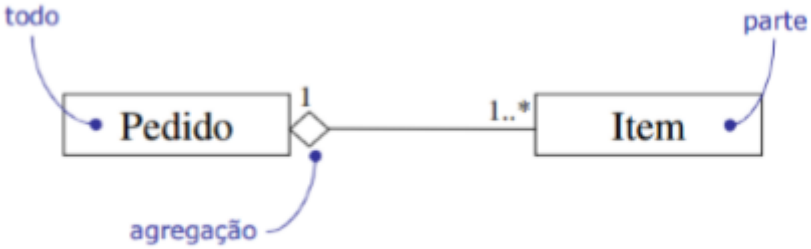
Relacionamentos: Nome | Sentido de Leitura | Navegabilidade (indicada por uma seta) | Multiplicidade (número de instâncias de objeto) | Tipo (Associação, Agregação, Composição, Generalização e Dependêncai) | Papéis.

Associação: é um relacionamento estrutural que indica que os objetos de uma classe estão vinculados com os objetos de outra classe. É representada por uma linha sólida conectando duas classes.

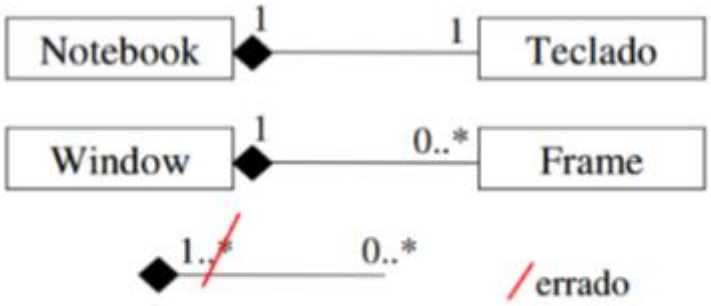
Indicadores de multiplicidade:

- 1 Exatamente um
- 1..* Um ou mais
- 0..* Zero ou mais (muitos)
- * mais (muitos)
- 0..1 Zero ou um
- m..n Faixa de valores (por exemplo: 4..7)

Agregação: é um tipo especial de associação utilizada para indicar "todo-parte"

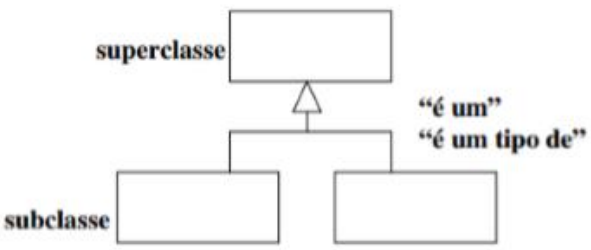


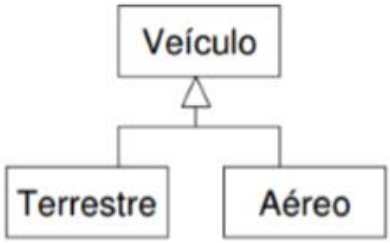
Composição: é uma variante semanticamente mais "forte" da agregação. Os objetos "parte" só podem pertencer a um único objeto "todo" e têm o seu tempo de vida coincidente com o dele.



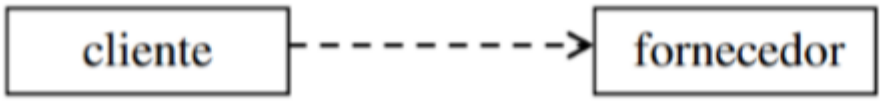
- Quando o "todo" *morre* todas as suas "partes" também *morrem*

Generalização: é um relacionamento entre itens gerais (superclasses) e itens mais específicos (subclasses).





Dependência: representa que a alteração de um objeto (o objeto independente) pode afetar outro objeto (o dependente)



Perspectivas:

- **Conceitos e Entidades:**
  - Representa os conceitos do domínio em estudo;
  - Perspectiva destinada ao **CLIENTE.**
- **Classes:**
  - Tem foco nas principais interfaces da arquitetura, nos principais métodos;
  - Perspectiva destinada aos **GERENTES DE PROJETO.**
- **Classes de Software:**
  - Aborda vários detalhes de implementação (navegabilidade, atributos etc);
  - Perspectiva destinada ao **TIME DE DESENVOLVIMENTO.**

DE Objeto:

- Está amplamente associado ao Diagrama de Classes;
- Fornece uma visão dos valores armazenados pelos objetos de um diagrama de classes em um determinado momento da execução de um processo de software.

DE Componentes:

- É uma peça básica na implementação de um sistema;
- Tipos:
  - Componentes de Instalação;
  - Componentes de Trabalho;
  - Componentes de Execução.
- Está amplamente associado à linguagem de programação que será utilizada para desenvolver o sistema modelado;

- Representa os componentes do sistema quando o mesmo for implementado.

DE Implantação:

- Determina as necessidades de hardware do sistema (todo o aparato físico).

DD Sequência:

- É um diagrama comportamental que se preocupa com a ordem temporal em que as mensagens são trocadas entre objetos;
- Baseia-se em um caso de uso e apoia-se no Diagrama de Classes para determinar os objetos das classes envolvidoas em um processo;
- Linhas Verticais: representam o tempo de vida um objeto;
- Linhas Horizontais ou Diagonais: representam mensagens trocadas entre objetos.

DD Colaborações:

- Está amplamente associado ao diagrama de sequência;
- Não se preocupa com a temporalidade do processo;
- Concentra-se em como os elementos do diagrama estão vinculados.

DD Estados:

- Um objeto possui um comportamento e um estado;
- O estado de um objeto depende da atividade na qual ele está processando;
- Mostra os possíveis estados de um objeto e as transações responsáveis pelas suas mudanças de estado;
- Pode basear-se em caso de uso.

DD Atividade:

- Preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica;
- Ilustra o fluxo de controle entre as atividades, enquanto que o DD Estados ilustra o fluxo de controle entre estados;
- Elementos:
    - Activity: Atividade propriamente dita;
    - Partition: É uma raia;
    - Decision: É uma estrutura de controle;
    - Initial: Primeiro elemento do diagrama;
    - Final: Último elemento do diagrama;
    - Fork: Tem como finalidade dividir o fluxo em mais de uma direção;
    - Join: Tem finalidade inversa, faz a união de várias direções em uma única.

LEGENDA:

Conceitos;

Vantagens / Desvantagens;

Exemplos;

Comentários;

Importância (Certeza que vai cair na prova);

AULA 15 – TESTE DE SOFTWARE

Três regras de Myers:

- 1º A atividade de teste é o processo de executar um programa com a intenção de descobrir um erro;
- 2º Um bom caso de teste é aquele que apresenta uma probabilidade de revelar um erro ainda não descoberto;
- 3º Um teste bem sucedido é aquele que revela um erro ainda não descoberto.

O OBJETIVO PRIMORDIAL DO TESTE É O DE **ENCONTRAR ERROS.**

Segundo Myers, quanto mais cedo descobrirmos e corrigirmos o erro, menor é o seu custo para o projeto.

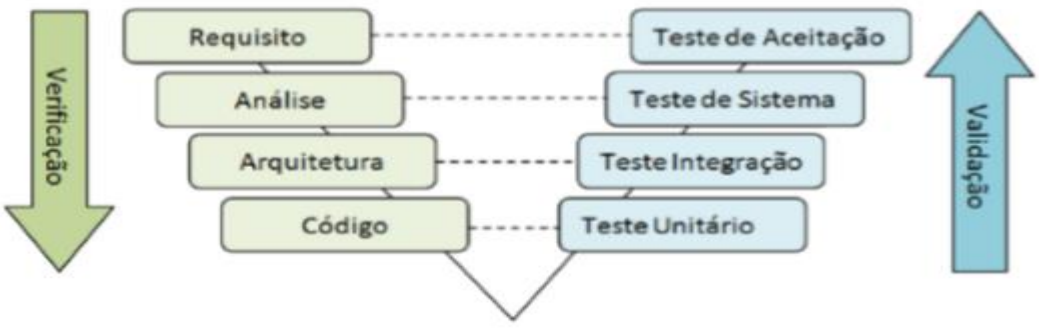
Teste de Caixa Preta: refere-se a todo teste que implica na verificação do funcionamento do software **através de suas interfaces**, o que, geralmente, permite verificar a operacionalidade de todas as suas funções. É importante também observar que como o software está organizado internamente não tem real importância.

Teste de Caixa Branca: em um produto de software está relacionado a um exame minucioso de sua estrutura interna e detalhes procedimentais.

- Não se deve guardar a falsa ideia de que o teste garantirá 100% do funcionamento do software para a correção final. Porque haverá um grande números de caminhos que o software pode oferecer.

Tipos de Teste:

Modelo "V": ~~ variação do modelo cascata ~~



Pergunta-se:

Verificação: “fizemos o software corretamente?” – Significa que os requisitos e funcionalidades documentados foram implementados!

Validação: “fizemos o software correto?” – Significa que os requisitos, independente do que foi planejado, estão sendo implementados para atender a regra de negócio do cliente, ou seja, se o sistema é realmente aquilo que o cliente quer e está pagando para ter.

Enquanto o enfoque do modelo cascata está nos documentos e nos artefatos, o V está na atividade e na correção.

Teste de Unidade:

- Objetifica a verificação de erros existentes nos módulos do projeto;
- Utiliza-se as informações contidas no documento de projeto detalhado do software;
- É uma técnica de teste de caixa branca.

Teste de Integração:

- Busca erros surgidos quando fazemos a integração das diferentes unidades e componentes do software;
- Os erros mais encontrados são os de interface devido as incompatibilidades de interface entre módulos que deverão trabalhar de forma cooperativa.

Teste de Validação:

- É a verificação de que o software como um todo cumpre corretamente a função para a qual ele foi específicado

**** Teste Alfa:**

- São realizados com base no envolvimento de um cliente, ou numa amostra de clientes, sendo realizado nas instalações do desenvolvedor do software;
- O desenvolvedor observa o comportamento do cliente e vai registrando as anomalias detectadas durante a utilização do software.

**** Teste Beta:**

- Voltado para softwares cuja distribuição atingirá grande número de usuários específicos de uma ou várias empresas compradoras;
- O desenvolvedor geralmente não está presente;
- Os clientes registram os problemas e relatam aos desenvolvedores em intervalos regulares de tempo.

Teste de Sistema:

- São verificados os aspectos de funcionamento do software, integrado aos demais elementos do sistema.

**** Teste de Recuperação:**

- O objetivo é observar a capacidade do sistema para recuperar-se da ocorrência de falhas, num tempo previamente determinado;
- As falhas são provocadas artificialmente.

**** Teste de Segurança:**

- Visa garantir que o sistema se comporte adequadamente diante as mais diversas tentativas ilegais de acesso.
- Exemplo: em um sistema escolar o aluno pode ver o histórico de notas nas disciplinas que cursou, mas não pode alterá-las.

**** Teste de Estresse:**

- Consiste em verificar como o sistema vai se comportar em situações limite;
- Exemplos: a) Limite em termos de quantidade de usuários conectados a um determinado sistema-servidor | b) Quantidade de utilização de memória e processador.

**** Teste de Desempenho:**

- Consiste em verificar se os requisitos de desempenho estão sendo atendidos para o sistema como um todo.

Casos de Teste: é um conjunto de condições usadas para teste de software

- Pode ser elaborado para indicar defeitos na estrutura interna por meio de situações que exercitem adequadamente todas as estruturas utilizadas na codificação.

Projeto de Testes: contempla a criação dos casos de teste e demais artefatos necessários as atividades de execução deles conforme definido no documento de Plano de Teste.

- Um caso de teste é realizado para cada caso de uso.