**Aplicações no Cotidiano**

**WEB**

World Wide Web designa um sistema de documentos em hipermídia que são interligados e executados na Internet. Os documentos podem estar na forma de vídeos, sons, hipertexto e imagens.

**ERP**

Enterprise Resource Planning ou Gestão Integrada de um Negócio. É um sistema de informação que interliga todos os dados e processos de uma organização em um único sistema.

**CRM**

Customer Relationship Management ou Gestão de Relacionamento com o Cliente. Software de gerenciamento da relação do negócio com o cliente. Satisfação de fidelização do cliente. Processos bem organizados e automatizados. Diminui gastos e elevam lucros e produtividade sustentável.

**Mobile**

Fácil acesso das informações. Serviço precisa ser atrativa com experiência única para o cliente. Desenvolvedores e manutenção para diferentes sistemas operacionais.

**IOT**

Internet das coisas é um conceito que se refere à interconexão digital de objetos cotidianos com a internet, conexão dos objetos mais do que das pessoas. Em outras palavras, a internet das coisas nada mais é que uma rede de objetos físicos capaz de reunir e de transmitir dados.

**IA**

Inteligência Artificial é a inteligência similar à humana exibida por sistemas de software, além de também ser uma área de estudo acadêmico.

**Processo de Desenvolvimento**

Um processo de software é um conjunto de atividades relacionadas que levam a produção de um sistema de software.

**ADIT**

As atividades do processo ADIT são: Análise, Design, Implementação e Testes.

***Análise***

As atividades de análise estão diretamente relacionadas com a descoberta de requisitos.

***Design***

É realizada a modelagem da solução de software, onde definimos a arquitetura e as especificações.

***Implementação***

É realizada a codificação do programa.

***Teste***

São realizadas as ações de validação e verificação. A partir de um plano de testes os requisitos são validados.

**Requisito**

Uma condição ou capacidade necessária para um usuário resolver um problema ou alcançar um objetivo.

Uma condição ou capacidade que deve ser atendida ou tida por um sistema ou componente do sistema para satisfazer a um contrato, padrão, especificação ou outro documento formalmente imposto.

Uma representação documentada de uma condição ou capacidade.

**Engenharia de Requisitos – Fases:**

**Elicitação**

Investigação, busca e descobertas dos requisitos.

**Análise**

Avaliação de possíveis conflitos, identificação das relações com o contexto, definição dos requisitos.

**Especificação**

Documentação e detalhamento das especificações dos requisitos.

**Validação**

Validação dos requisitos em relação aos propósitos do produto de software.

**Crise do Software**

A “crise do software” foi um termo cunhado para descrever as dificuldades enfrentadas no desenvolvimento de software no fim da década de 60. A complexidade dos problemas, a ausência de técnicas bem estabelecidas e a crescente demanda por novas aplicações começavam a se tornar um problema sério.

**Ciclo de Vida Tradicional:**

**Cascata – Waterfall (Royce)**

Particionamento inflexível. Para requisitos definidos e mudanças limitadas. Mais utilizado para projetos pequenos.

**Incremental (Mills)**

Valor a cada incremento. O incremento inicial funciona como um protótipo para auxiliar na elicitação dos requisitos. Cada incremento é uma parte do sistema.

**Espiral (Bohen)**

Definição de objetivos. Avaliação e redução de riscos. Desenvolvimento e validação. Planejamento.

**RUP – Rational Unified Process**

Fases: Iniciação; Elaboração; Construção e Transição.

Disciplinas: Projeto; Gestão.

Esse é o modelo mais completo entre todos, mas que não deve ser usado para qualquer tipo de projeto.

**Ciclo de Vida Ágil:**

**Manifesto Ágil**

Satisfação do cliente entrega contínua / Mudanças dos requisitos são bem vindas / Entrega frequente funcionando / Cliente e desenvolvedor trabalhando em conjunto / Indivíduos motivados e com confiança / Reuniões face a face / Software funcionando como medida de processo / Desenvolvimento sustentável / Excelência técnica e bom design contínuo / Simplicidade é essencial / Equipes auto-organizáveis / Refinamento e ajustes para ser eficaz.

**Para o professor, existem 3 tipos de projetos:**

Projeto Hobbie; Projeto Pequeno e Projeto Colossal

**SCRUM**

Foco na equipe. Melhoria contínua. Equipes enxutas. Definição de backlog. Definição de Sprint.

**Lean**

Valor no ponto de vista cliente final. Fluxo de valor para cada família de produto. Criação de valor para o produto fluir para o cliente. Deixe o cliente extrair valor da atividade. Achar o valor perfeito sem desperdício.

**Kanban**

Termo de origem japonesa para “cartão” ou “sinalização”. Usado em conjunto com o Scrum.

**Modelo de Software**

**Abstração**

Processo de destaque de aspectos relevantes e omissão daqueles irrelevantes para um certo propósito.

**Modelo**

Representa aspectos relevantes de um mundo de acordo com um certo propósito.

**Erros de Abstração**

A abstração é uma atividade realizada via um processo mental, que ocorre na mente do analista, e podemos cometer erros de interpretação.

Existem 4: Elemento Omisso; Elemento Detalhe; Elemento Estranho; Elemento Alienígena

**Modelagem de Domínio**

É uma atividade para identificar os termos utilizados no projeto do software ou para construir o glossário do projeto. O modelo de domínio define os fundamentos de escopo e elementos para construção do uso do software. Oferece um vocabulário comum para a comunicação entre membros de equipe.

**Diagrama de Classe de Domínio**

O DCL (Diagrama de Classe) é um diagrama da UML, que usaremos neste contexto como ferramenta para desenhar o DOMÍNIO.

**Interface com o Ambiente:**

**Objetivos**

Criar um esboço de interface com o cliente. Ajudar na identificação dos comportamentos de uma aplicação. Estruturar o fluxo de dados e o fluxo de controle.

**Fluxo de Dados**

Entrada > Saída > Label > Lista > Combo Box > Radio > Check > Janela.

Caso tenha interesse em fazer, observar o utilizado pelo professor na aula 04 – Interface com o Ambiente.

**Fluxo de Controle**

Ação > Continua/Cancela > Adiciona/Remove.

**Tipos de Dados:**

**Dado**

É a representação de uma grandeza ou característica de algum elemento.

**Tipos de Dados (Básicos):**

Numéricos > Naturais; Inteiros; Reais.

Caracteres > Alfabéticos.

Cadeia de Caracteres > Alfabéticos (Palavra ou Frase).

Lógicos > Verdade; Falso.

**Abstração dos Dados**

Envolve a organização dos dados de um problema em representações estruturas e significativas para o negócio.

**O Caso IMC:**

**IMC – Índice de Massa Corporal**

Para saber o peso ideal, é utilizado o IMC. O método foi criado por Lambert Quételet no fim do século XIX.

Regra para o cálculo: IMC = massa/(altura x altura) ou massa dividido por altura vezes altura.

**FURPS+**

**Requisitos Funcionais**

Associados às funcionalidades do sistema e aos resultados de comportamento do sistema. Requisitos que resolve as necessidades do usuário final. Requisitos funcionais de usuário e requisitos funcionais do sistema.

**Requisitos Funcionais Regras de Negócio**

Requisitos funcionais relacionados com o que o sistema deve fazer. Regra de negócio relacionadas com como o sistema deve fazer. As regras de negócio relacionadas com a premissas observadas em um negócio aconteça. As necessidades necessitam de focos diferentes tanto para os requisitos funcionais como para as regras de negócio.

**Requisitos não Funcionais Usabilidade**

Facilidade de aprendizagem. Eficiência da tarefa. Facilidade de recordação. Entendimento. Satisfação subjetiva.

**Requisitos não Funcionais Performance**

Tempo de resposta. Taxa de transferência. Capacidade. Partida. Parada.

**Requisitos não Funcionais Segurança**

Associados à política de segurança do negócio. Procedimentos para que o sistema permaneça operante mesmo com acesso indevido. Validação de dados de entrada no sistema. Registro de logs dos eventos para auditoria e análise. Proteção exigida para a segurança do sistema.

**Requisitos não Funcionais Confiabilidade**

Exatidão. Disponibilidade. Recuperabilidade. Frequência e severidade das falhas.

**Fases**

**Elicitação**

Nesta fase são realizadas as atividades relacionadas a descoberta dos requisitos e compreensão do que deve ser feito. A elicitação não esgota toda a extensão dos requisitos, mas permite a descoberta ao longo do projeto.

**Stackeholders**

São pessoas ou organizações que serão afetadas pelo sistema e que tem influência direta ou indireta nos requisitos do sistema. Normalmente são os responsáveis pela tomada de decisão e aprovação do projeto.

Exemplos: Dono da empresa, Gerentes, Especialistas de Domínio.

**Técnicas de Elicitação**

Interativas: Entrevista, Questionário, Observação, Análise de Processo de Negócio.

Descritivas: Modelo de Domínio, Análise de Documentos, Modelo de Caso de Uso, História do Uso.

**Lista de Requisitos**

Resultado dessa fase é uma lista de requisitos ou diagramas para ilustrar o entendimento das necessidades e metas dos usuários.

**Análise**

Analisar as características de cada requisito, garantindo que será desenvolvido quando estiver completo, estável e aprovado pelo usuário.

**Características de um requisito bem definido**

Necessário; Objetivo; Correto; Completo; Consistente; Não-ambíguo; Compreensível; Modificável; Verificação; Rastreável.

**Atributos de Requisitos**

Prioridade; Complexidade; Completude; Dificuldade; Estabilidade; Risco.

**Criar Critérios para os Atributos**

Os critérios podem ser definidos em diversos tipos de métricas:

1. [1..10]
2. [baixa, média, alta]
3. [completo, incompleto]

**Especificação**

É a etapa dedicada a representar os requisitos de uma forma que eles possam perdurar ao longo do tempo e possam ser verificados e validados posteriormente. Isso pode implicar em formatos diferentes de especificação que envolvem textos, diagramas e tabelas.

**Notações de Escrita de Requisitos**

Sentenças de linguagem natural – É a linguagem falada. Cada frase deve expressar um requisito.

Linguagem natural estruturada – São escritos em um formulário ou template. Cada campo fornece informações sobre um aspecto do requisito.

Notações gráficas – Modelos gráficos, suplementados por anotações em texto, são utilizados para definir os requisitos funcionais do sistema. São utilizados com frequência os diagramas de casos de uso e de sequência da UML.

Especificações matemáticas – Essas notações se baseiam em conceitos matemáticos como as máquinas de estados finitos ou conjuntos. Embora essas especificações inequívocas possam reduzir a ambiguidade em um documento de requisitos, a maioria dos clientes não compreende uma especificação formal. Eles não conseguem averiguar se ela representa o que desejam e relutam em aceitar essa especificação como um contrato do sistema (discutirei essa abordagem no Capítulo 10, que aborda a dependabilidade do sistema).

**Comunicação**

Todo artefato de interação com o usuário deve ser “devidamente” comunicado (em tempo adequado, forma, estrutura, conteúdo, meio, frequência, granularidade).

Quanto mais simples e padronizado forem os artefatos que o usuário tiver que aprovar, mais ágil deveria ser o processo de levantamento, especificação e aprovação.

Quanto mais preciso forem estes artefatos, menor retrabalho será realizado em função de baixa qualidade dos requisitos. Toda comunicação deve ser realizada conforme o plano de comunicação definido para o projeto.

**Elicitação Interativa**

**Entrevista Estruturada**

Entrevista estruturada ou fechada. Perguntas criadas para serem respondidas de forma direta. Perguntas objetivas, o entrevistado não tem espaço para explicar as respostas. Perguntas alinhadas com os requisitos levantados para investigação e esclarecimento. Verdadeiro ou falso. Marcar alternativas em questionário.

**Entrevista Não Estruturada**

Entrevista não estruturada ou entrevista aberta. Poucas perguntas para elucidar requisitos. A sequência das perguntas pode ser alterada de acordo com as respostas do entrevistado. Entrevistado responde de forma livre e aberta, sem interferências. O entrevistado é um ouvinte, apenas faz anotações. É basicamente o oposto da entrevista estruturada.

**Entrevista Semiestruturada**

É um pouco de estruturada e não estruturada. Perguntas abertas e fechadas, com respostas objetivas e subjetivas.

Um misto de perguntas para elucidas requisitos. O entrevistador fica atendo às respostas, anotando o comportamento do entrevistado.

**Questionário**

Busca informações sobre o negócio e a interação com o usuário final que está distante, ou necessidade de volume de informações. Perguntas abertas ou fechadas com sequência lógica. Questionário é frio e impessoal, com resultados pouco ricos que podem ser utilizados para tratamento estatístico. As respostas dão um direcionamento para a análise dos requisitos.

**Observação**

Utilizada em aplicações em campo. Entender de perto sobre o negócio e as necessidades que precisam ser resolvidas. Além de ver e ouvir, há a percepção de situações específicas e fundamentais para o negócio. Sem interferir o ambiente, mesmo que crie curiosidade e desconforto. O período da observação vai depender de se obter os resultados desejados.

**Processo de Negócio**

Implementam as estratégias, metas e objetivos corporativos. Viabilizam pela infraestrutura corporativa: Estrutura Organizacional (Organograma), Sistema de informação, Infraestrutura tecnológica e física. Essência do que a organização é e faz no dia a dia. Alinhados com estratégias, metas e objetivos corporativos. Infraestrutura precisa está alinhada.

**Modelo de Domínio**

É uma atividade para identificar os termos utilizados ou para construir o glossário do projeto.

Para desenhar o diagrama de classe de domínio, vamos utilizar o programa: <https://app.diagrams.net> (draw.io)

**Coleta de Documentos**

Os documentos da área podem ser previamente coletados para estudo e preparação das entrevistas.

Exemplos: Formulários, Relatórios, Manual do usuário, Descrição de procedimentos.

Vantagens: Identificação dos documentos pertinentes ao processo, Coleta de dados quantitativos, encontrando taxas de erro em documentos existentes ou inadequados, Promove condições para análise e refinamento do processo.

Desvantagens: Dispersão das informações e volume do trabalho, Documentação pode estar desatualizada, Redundância.

**Casos de Uso (Cenário)**

São narrativas em texto, amplamente utilizadas para descobrir e registrar requisitos. Eles influenciam muitos aspectos de um projeto. Informalmente, casos de uso são narrativas em texto de algum ator usando um sistema para atingir objetivos.

**Cenário**

É uma sequência específica de ações e interações entre atores e o sistema. É uma história particular de uso de um sistema ou um caminho através do caso de uso.

**Estrutura de Caso de Uso**

UC# (Use Case – N°); Descrição; Atores; Pré-condição; Fluxo básico; Fluxos alternativos; Fluxos de exceção; Pós-condição.

**História de Usuário (Cenários)**

É uma ferramenta de elicitação de requisitos baseada na descrição de histórias de usuário (user stories) para o entendimento das necessidades e metas dos usuários.

Definição: É uma descrição concisa de uma necessidade do usuário do produto (ou seja, de um “requisito”) sob o ponto de vista desse usuário. A User Story busca descrever essa necessidade de uma forma simples e leve.

Elementos: Cartão, Conversa e Confirmação.

**Cartão**

A ideia do cartão é garantir que a descrição da História seja sucinta, objetiva e direta. Contamos a história do usuário usando o formato “Quem”, “O quê”, e “Por que”.

Literalmente um cartão, como um cartão de visita.

**Conversas**

Traz o envolvimento do cliente para definir critérios e testes de aceitação. Compreensão compartilhada. Agregar valor ao negócio e retorno ao investimento.

Envolvendo os desenvolvedores e clientes para discussão e escrita dos critérios de aceitação. – *Só os condôminos podem usar o App*.

**Confirmação**

São critérios e testes deles derivados que documentam os detalhes da User Story, definindo seus limites. Ou seja, são regras que estabelecem como a funcionalidade deve se comportar uma vez implementada. Esses critérios são chamados de Critérios de Aceitação e os testes, de Testes de Aceitação.

Critério: o locatário deve ser do condomínio.

Teste de aceitação:

- o locatário é do condomínio

permitiu = correto não permitiu = errado, deve ser corrigido

- o locatário não é do condomínio

não permitiu = correto permitiu = errado, deve ser corrigido

**Elicitação Design Thinking Imersão Prelimiar**

**Imersão**

É a primeira fase do design thinking. Equipe entra em contato com o contexto do problema a ser resolvido. A fase de imersão considera o ponto de vista do cliente e o ponto de vista do usuário final. Imersão preliminar e a imersão em profundidade.

**Imersão Preliminar**

Primeira fase do design thinking. Ferramentas: reenquadramento, pesquisa exploratória e pesquisa de mesa. Entendimento da necessidade, dores e desejos do público-alvo. Alinhamento estratégico com a equipe de desenvolvimento, de negócios, do cliente e do usuário final.

Análise do comportamento do mercado, da sociedade das pessoas e do próprio negócio. Análise das tendências de mercado. Análise de comportamentos extremos, internos e externos.

**Reenquadramento**

Buscar observar os problemas que precisam ser inovados ou melhorados. Examina e observa o negócio de várias perspectivas. Desconstrução de objeções e crenças e achismos do negócio. Quebra de paradigmas. Permite alterações no pensamento em relação ao negócio.

Abertura de novos caminhos para soluções inovadoras. Soluções observadas sob um novo olhar. Um dia de workshop ou encontros diários.

**Pesquisa Exploratória**

Informações do público-alvo: comportamentos e vivências. Conexão da nova realidade do produto ou serviço. Investigação das necessidades, dores e desejos, além do comportamento do público-alvo.

**Pesquisa de Mesa**

Busca de dados e informações da solução. Artigos, revistas, jornais e internet. Informações divergentes do usuário-final. Tendências internas e externas sobre o negócio.

**Elicitação Design Thinking Imersão em Profundidade**

**Imersão em Profundidade**

Objetivo de observar o ser humano, seu comportamento. O que está falando, agindo, pensando e sentindo. Empatia para o entendimento das necessidades, dores e desejos.

**Entrevistas**

Busca de informações por meio de perguntas e respostas. Entendimento da vida e comportamento dos entrevistados. Busca de informações das experiências e vida do entrevistado. Busca da motivação pelas respostas apresentadas.

Importância da análise das expressões faciais e corporais. Importância da análise do tom de voz e gestos. As expressões faciais e corporais devem ser consideradas. Local neutro como as ruas ou no ambiente familiar ou do trabalho.

**Caderno de Sensibilização**

Objetivo de observar as informações do público-alvo e como é seu dia a dia. Entender o comportamento do público-alvo. Entender como uma ação ou atividade pode interferir no cotidiano, no ambiente em que convive e as pessoas afetadas. Interferências dessas informações no propósito do cliente final.

Registro dos resultados relacionados ao usuário final. Resultados utilizados em outras ferramentas do design thinking.

**Sessão Generativa**

Objetivo de realizar atividades presenciais com os usuários finais. Análise das experiências observadas. Apresentação das visões e experiências do dia a dia dos clientes finais. Análise do sentimento e das emoções do cliente final.

Fazer o usuário final experimentar as possibilidades das soluções do negócio. Utilizam as informações do caderno de sensibilização. Buscam por memórias, sentimentos e motivações da experiência do usuário final.

**Um dia na vida**

Objetivo de representar um dia do usuário final. Entendimento e análise do comportamento e interesses do usuário final. Equipe e desenvolvimento e de negócios assumem os papéis do usuário final e dos outros atores envolvidos no negócio. Simulação de um dia do usuário sob o ponto de vista com um diferente olhar.

Conhecer os detalhes do dia a dia do usuário final. Ações, comportamentos e emoções são analisados. Simulação do cotidiano do usuário final.

**Sombra**

Alguém da equipe vai acompanhar a interação do usuário final com o produto ou serviço. Atuação como sombra penas observando sem interagir ou intervir. Obter informações importantes para o negócio. Entendimento mais rico sobre comportamento, emoções, sentimentos, nas expectativas e hábitos do usuário final.

**Análise**

**Tipos de classificação**

- Corretude

todos são significativos?

- Completos

todos são significados?

- Consistentes

entre si.

- Não-ambíguos

ter apenas um único sentido

- Verificáveis

testáveis

- Rastreáveis

origem

relacionamento

**Priorização**

Priorização de utilização dos recursos. Priorização de requisitos pois os recursos são limitados. Precisa escolher o que será desenvolvido primeiro e o que será desenvolvido depois. A prioridade otimiza os recursos e diminui os custos.

**Complexidade**

Alguns requisitos devem ter um nível de complexidade. Os requisitos de complexidade devem estar alinhados com as estratégias do negócio. Senhas, por exemplo, devem atender os requisitos de complexidade por motivo de segurança.

**Completude**

Os documentos de requisitos devem ser completos. Deve conter todos os requisitos relevantes. Cada requisito deve ser documentado de forma completa. A engenharia de requisitos precisa determinar métricas para ter a completude do documento de requisito considerando tempo e recursos disponíveis.

**Matrizes de Requisitos**

Matriz de requisitos ou matriz de rastreabilidade de requisitos. Ferramenta para gerenciamento das demandas de um projeto. A matriz de requisitos deve ser desenvolvida assim que os requisitos forem levantados. Serve de documentação dos requisitos. Garante o alinhamento dos objetivos do projeto e do negócio na entrega. É um documento.

**Técnicas de Levantamento de Requisitos**

**Casos de Uso Fluxo Básico**

Descreve o cenário de sucesso de uma interação do usuário com o sistema. Também conhecido como cenário “dia feliz”. O Usuário atinge o objetivo do cenário.

**Casos de Uso Fluxos Alternativos**

Descreve um ou mais cenários que vão oferecer alternativas de interação entre o usuário e o sistema, a partir de um passo do fluxo básico para que o usuário tenha sucesso no cenário.

**Casos de Uso Fluxos de Exceção**

Descreve um ou mais cenários em que o fluxo de interações impede o ator de atingir o objetivo do cenário. “Caso de insucesso”.

**Glossário**

É um tipo de dicionário específico para palavras e expressões pouco conhecidas, seja por serem de natureza técnica, regional ou de outro idioma.

**Glossário nos Requisitos**

Define o vocabulário comum sobre o sistema em desenvolvimento e é uma importante peça para melhorar a comunicação entre os membros da equipe. “Pode ser feito a partir do modelo de domínio.”

**Dicionário de Dados**

É uma coleção de metadados que contém definições e representações de elementos de dados.

**Modelo de Dicionários de Dados**

Nome / Tipo de Dado / Descrição / Limite inferior / Limite superior.

**Validação**

**Validação Requisito Usuário**

Entendimento do que o usuário realmente deseja. Certificar que não houve falha de comunicação com o usuário. Demonstrar que os requisitos definem o sistema que o usuário realmente deseja. Diminui custos com erro de implementação. Técnicas: revisões de requisitos

prototipação

geração de casos de teste

**Validação Requisito Envolvidos (Stakeholders)**

A validação de um requisito de envolvidos, também conhecidos como stakeholders, tem o objetivo de descobrir possíveis erros que foram elicitados e documentados. Quando utilizada, é possível demonstrar que os requisitos definem o sistema que o envolvido realmente deseja, diminuindo os possíveis custos com os erros de implementação. Técnicas: revisões de requisitos

prototipação

geração de casos de teste

**Validação Requisito Técnico (Sistema)**

Descrições mais detalhadas dos requisitos de usuário. Entendimento do que é necessário tecnicamente. Certificar que não houve falha de comunicação com o sistema. Demonstrar que os requisitos definem que o sistema tenha condições de funcionar adequadamente. Diminui custos com erro de implementação. Especificação completa e consistente de todo o sistema.

A validação de um requisito técnico tem o objetivo de descobrir possíveis erros que foram elicitados e documentados tecnicamente. Durante a validação dos requisitos dos técnicos, é importante validar que o sistema realmente precisa entregar que satisfaçam as necessidades do usuário final. Neste caso, é importante entender o que é necessário tecnicamente. Quando utilizada, é possível demonstrar que os requisitos definem o sistema que o envolvido realmente deseja, diminuindo os possíveis custos com os erros de implementação.

**Verificação Requisito Usuário**

A verificação certifica se os requisitos funcionais e não funcionais dos usuários são atendidos. Validação e verificação dos requisitos de usuário são realizados em conjunto e são dependentes. Objetivo de eliminar falhas e fatores que podem prejudicar a qualidade do produto ou serviço. Redução de riscos e controle de qualidade. Testes estáticos evita retrabalho e atrasos no cronograma e custos extras.

A verificação de requisitos é uma forma científica de verificar se os requisitos funcionais e não funcionais, neste caso, os requisitos de usuário, são atendidos. É realizada em conjunto com a validação de requisitos de usuário e são dependentes. É possível reduzir e diminuir os riscos e ter um maior controle de qualidade.

**Verificação Requisito Envolvido**

A verificação corresponde ao conjunto de atividades de controle de qualidade. Validação e verificação dos requisitos de envolvidos são realizados em conjunto e são dependentes. Objetivo de eliminar falhas e fatores que podem prejudicar a qualidade do produto ou serviço. Garantia da construção do produto certo. Testes estáticos evita retrabalho e atrasos no cronograma e custos extras.

**Verificação Requisito Técnico**

A verificação corresponde ao conjunto de atividades de controle de qualidade, neste caso, tecnicamente ao sistema. É importante verificar se os requisitos estão claros, corretos e completos. Objetivo de eliminar falhas e fatores que podem prejudicar a qualidade do produto ou serviço. Particularidades técnicas e qualidades globais são verificadas: facilidade de uso, disponibilidade, portabilidade, integridade, eficiência. Testes estáticos evita retrabalho e atrasos no cronograma e custos extras.

**Diagrama**

**Diagrama de Casos de Uso**

Diagrama que representa o sistema pelo olhar do usuário final. Especifica e documenta requisitos. Observação das principais funcionalidades do sistema e suas interações com o usuário final. Elementos: cenário, ator, caso de uso, comunicação.

**Extensões do Diagrama de Caso de Uso**

Relacionamentos entre casos de uso: inclusão – include, extensão – extend,

generalização ou herança – generalization.

**Diagrama de Atividades**

Diagrama comportamental que especifica o comportamento do software.

Diagrama que representa: o funcionamento do software em nível macro e micro.

a execução das partes.

a atuação do sistema em relação ao negócio.

Semelhante a um fluxograma. Objetivo de especificar o que é o software e para que serve o software. Serve de documentação dos requisitos funcionais com o fluxo da informação.

**Diagrama de Classes**

Representação da estrutura e ralações das classes. As classes são modelos para objetos. Conjuntos de objetos com as mesmas características agrupados. As classes representam tabelas no banco de dados. É importante considerar um cenário para desenvolver o diagrama de classes. Abstração das classes para efetuar ligação e cardinalidade.

**Diagrama BPMN**

BPMN – Business Process Modeling Notation ou Notação de Modelagem de Processos de Negócios.

Objeto de Fluxo: definem o comportamento do processo de negócio (eventos, atividades e decisões).

Objeto de Conexão: forma como os objetos de fluxo se conectam (fluxo de sequência, fluxo de mensagem, associação).

Raia da Piscina: forma de organização das atividades em categorias (pool, lane).

Artefatos: utilizados para informações adicionais no processo (objeto de dados, grupo, anotações).

**Revisão e Inspeção**

**Qualidade de Software**

É difícil escrever requisitos de software completos e inequívoco. Os desenvolvedores de softwares e os clientes podem interpretar os requisitos de maneiras diferentes, e pode ser impossível chegar a um acordo sobre se o software está em conformidade com a sua especificação ou não.

**Revisão**

É uma forma para examinar o seu próprio trabalho.

Numa revisão pessoal: o analista revisa seu produto sozinho

o objetivo é encontrar defeitos antes dos testes

revisões são mais eficazes quando estruturadas e medidas

**Revisão de Requisitos**

Na engenharia de requisitos, em cada fase é gerado um conjunto de artefatos.

Elicitação – lista de requisitos.

Análise – priorização dos requisitos.

Especificação – casos de uso, histórias dos usuários, diagramas.

Validação – contrato.

**O que revisar?**

Aderência entre os NM dos usuários e os artefatos gerados. O formato e padronização dos artefatos. Nível de detalhamento adequado. Os pontos de verificação.

**Defeito, Erro e Falha**

Defeito: São imperfeições encontradas no software ou em seus processos de levantamento de requisitos.

Erro: São imperfeições encontradas no código em decorrência de defeitos não resolvidos no levantamento de requisitos.

Falha: É a materialização (ocorrência) do erro encontrado no código.

**Revisão do Código**

Requisitos; Interpretação incorreta dos requisitos; Erros em algoritmos; Diferenças entre produzido com especificado.

**Testes**

O teste de software mostra as falhas do sistema antes que o desenvolvimento seja concluído. A partir desse teste, é possível assegurar que as funcionalidades solicitadas estejam presentes e de acordo com o esperado.

**Processo ADIT e Testes**

Fase de Análise do ADIT permite a criação dos planos de testes a partir dos requisitos. O artefato gerado é o plano de testes.

**Inspeção**

São revisões por pares, em que os membros do time colaboram para encontrar defeitos e erros nos artefatos.

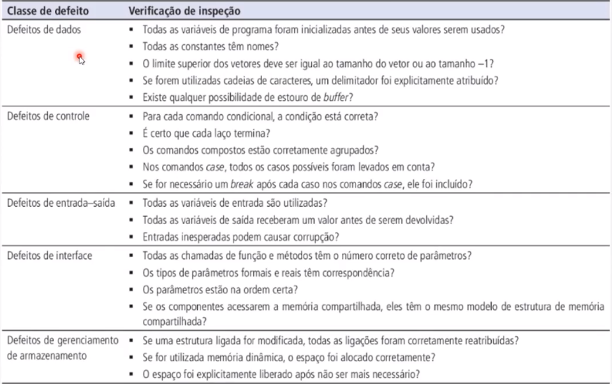
**Inspeção de Requisitos**

Visa encontrar os defeitos ligados aos artefatos gerados na engenharia de requisitos.

**Inspeção de Código**

No ADIT estaremos na fase de Implementação, os requisitos são usados para realizar a inspeção do código. Quem produz o código não inspeciona o próprio código.

**Checklist de Inspeção**



**Inspeção de Testes**

O artefato de plano de testes é elaborado ainda na fase de análise. O plano de testes reflete os requisitos, desta forma a inspeção reforça a qualidade dos requisitos.

**Momentos de Inspeção de Testes**

Na lista de requisitos, pois uma característica de um requisito de boa qualidade é ser testável. No momento de desenvolvimento, quando usada programação em pares. No plano de testes gerado, pois verifica se o plano está aderente aos requisitos.