**uSJT – Universidade São Judas Tadeu**

**ATIVIDADE 06 – GRUPO 4**

**UC : Modelagem de *software***

**Professor : Edgard Luiz Bernardes Valderramas**

**Semestre : 1º Semestre - 2024**

A Atividade 06 é composta por quatro partes:

Parte I - Pesquisa

Parte II – Exercícios de fixação

Parte III – Exercícios discursivos

Parte IV - Desenvolvimento

**PARTE I – Pesquisa**

Realizar uma pesquisa sobre quatro tipos de modelos de processos de desenvolvimento de software do mercado, citando:

1. origem;
2. características;
3. aplicabilidade.

**1 - Modelo RAD**

O modelo RAD teve sua origem nos finais dos anos 1980 até 1990, ele foi desenvolvido para ser oposto as abordagens tradicionais que já existiam que muitas vezes eram vistas como lentas e inflexíveis para atender as necessidades dos clientes. O conceito dele era entregar modelos mais ágeis e funcionais que forma rápida, usando método iterativos e incrementais.

Suas características são: Ela possui desenvolvimento incremental que divide o projeto em incrementos funcionais pequenos e entregáveis. Participação ativa do cliente, podendo fornecer feedbacks positivos ou negativos para o desenvolvimento. Ele reutiliza componentes e ferramentas para softwares existentes que aceleram o processo de construção de sistemas, ajudando a reduzir o tempo e riscos. Além de possuir ciclos de desenvolvimentos curtos nos quais os requisitos são identificados, implementados e testados em um curto período de tempo, geralmente de algumas semanas a alguns meses. Permitindo uma entrega rápida de valor aos clientes e uma resposta ágil a mudanças nos requisitos.

Sua aplicabilidade pode ser usada em projetos voláteis, curtos e que possuam uma pressão por entrega rápida, permitindo ser uma escolha adequada. Ele permite que o software seja desenvolvido e entregue em incrementos pequenos e funcionais em um curto período de tempo. Pode ser também para projetos de tamanho médio a grande sendo frequentemente usado, onde a complexidade e a escala do software exigem uma abordagem ágil e iterativa de desenvolvimento.

**2 - Modelo Incremental**

O modelo incremental surgiu em meados dos anos 1980 entre 1990, surgindo como um modelo de resposta as limitações nos modelos tradicionais da época, como o modelo cascata por exemplo.

As características dela possuem: A divisão de incrementos na qual divide o projeto em partes menores e mais fáceis de ser gerenciadas, dando funcionalidade ao sistema. Possui também entrega continua de funcionalidades e o feedback de clientes, possibilitando que o software seja ajustado de acordo com as preferências dos clientes de acordo com os gostos deles. Além disso a flexibilidade e adaptabilidade também é permitido nesse modelo.

Sua aplicabilidade pode ser em projetos de longo prazo, possibilitando a entrega em pequenas partes. Possibilita a redução de riscos que podem impactar no projeto e também prioriza os recursos.

**3 - Modelo em Espiral**

O modelo em espiral, criado por Barry Boehm em 1988, é uma melhoria do Modelo Incremental e possui esse nome por causa de sua representação, onde cada volta no espiral percorre todas as fases do processo de software. Essa metodologia incorpora elementos de desenvolvimento iterativo e análise de riscos ao longo do tempo. O projeto é dividido em ciclos, cada um composto por quatro etapas: identificação de objetivos, análise de riscos, desenvolvimento e planejamento da próxima iteração. As principais vantagens do modelo em espiral estão na análise proativa de riscos, flexibilidade e prototipação funcional em cada ciclo. Além disso, ele não faz distinção entre desenvolvimento e manutenção, permitindo uma abordagem holística ao longo do ciclo de vida do software. No entanto, é importante considerar as necessidades específicas do projeto ao escolher a metodologia de desenvolvimento mais adequada

**4 - Modelo Cascata**

O modelo cascata, também conhecido como processo Waterfall, é uma metodologia de desenvolvimento de software linear e sequencial que surgiu em meados dos anos 70. Suas principais características incluem fluidez, correlação entre as etapas de desenvolvimento, divisão das tarefas em etapas predeterminadas e a necessidade de finalizar todas as tarefas de uma etapa antes de passar para a seguinte. No entanto, o modelo cascata não permite erros ou alterações, e se as próximas etapas não se desenvolverem adequadamente, todo o projeto deverá ser refeito. Ele é mais adequado para projetos pequenos com requisitos bem definidos e ambientes onde mudanças são raras e os requisitos permanecem estáveis. Em resumo, o modelo cascata é eficaz quando os requisitos são claros e não há necessidade de ajustes frequentes, mas é importante considerar suas limitações ao escolher a metodologia de desenvolvimento mais adequada para cada contexto

**PARTE II – Exercícios de fixação**

1. **Considere, por hipótese, que uma Analista participa do levantamento de requisitos de um sistema de controle de processos judiciários. Uma das funcionalidades do sistema realiza uma consulta e mostra a quantidade de processos por período nos diversos juizados do Maranhão; como entrada, solicita a data inicial e final e, como resultado, apresenta o Ano, o Mês, o Nome do Juizado, a Situação dos Processos e a Quantidade de Processos. Neste contexto,**
2. "o tempo de resposta da consulta não deve ultrapassar 5 segundos" é um requisito funcional de desempenho.
3. "o sistema deve ser desenvolvido em Java, integrado à plataforma Oracle e fazer interface com a API Twitter4Beans" é um requisito não funcional de usabilidade.
4. um requisito funcional deve atender ao atributo de Consistência. Este atributo indica que, quando o requisito assumir mais de uma responsabilidade, deve ser decomposto.
5. **um Caso de Uso pode especificar o comportamento de uma funcionalidade do sistema e a identificação dos atores pode se iniciar com os primários e prosseguir com os secundários, que oferecem suporte ao sistema para que os primários possam realizar seu trabalho.**
6. várias técnicas podem ser utilizadas, como o JAD – Joint Application Design, em que é realizada uma observação direta das atividades realizadas durante um período de trabalho de um funcionário visando encontrar requisitos que não seriam observáveis usando técnicas convencionais.
7. **Na análise de requisitos de um sistema negocial de operações demandado por um órgão público, foram relacionados os seguintes requisitos:**

I o sistema deve ter versões disponíveis para plataformas web e móvel (Android e iOS);

II o sistema deve restringir o acesso ao painel de gestão estratégica do sistema apenas a diretores do órgão;

III o sistema deve permitir que o painel de gestão estratégica, acessado pelos diretores, seja atualizado com os dados das operações negociais do órgão, a cada três minutos;

IV o sistema deve permitir que o relatório de fechamento mensal das operações seja disponibilizado aos diretores no primeiro dia útil do mês subsequente, via painel de gestão estratégica.

São requisitos funcionais desse sistema negocial apenas os itens

1. I e II.
2. I e III.
3. III e IV.
4. I, II e IV
5. **II, III e IV.**
6. **Suponha que um Analista de TI, participando da etapa de análise de requisitos de um sistema de emissão de certidão negativa para o TRF4, tenha elencado os requisitos apresentados abaixo:**

Utilizar interface responsiva para que possa ser executado em dispositivos móveis e na web.

1. Validar o tipo de certidão solicitado.
2. Emitir certidão negativa após verificação de situação do requerente.
3. Solicitar o CPF do requerente.
4. Responder ao clique único do usuário em qualquer botão da interface.
5. Validar o CPF do requerente.
6. Restaurar os dados automaticamente após falhas não programadas.
7. Solicitar o nome do requerente.
8. Oferecer dois tipos de certidão: para fins gerais e para fins eleitorais.
9. Emitir aviso de impossibilidade de emissão da certidão.

Sobre os requisitos, é correto afirmar que

1. todos são funcionais.
2. todos são não funcionais.
3. 1, 5 e 7 são não funcionais.
4. **apenas 3, 4, 8, 9 e 10 são funcionais.**
5. apenas 2, 6 e 7 são não funcionais.
6. **Um profissional da área administrativa de certa instituição recebeu um Analista de Sistemas que estava fazendo o levantamento de requisitos para a construção de um novo software. Ao informar ao Analista um requisito não funcional para seu departamento, o profissional corretamente disse que**
7. **a resposta a uma consulta de dados deveria durar no máximo dois segundos para não atrasar seu trabalho.**
8. o sistema deveria permitir a alteração de dados incluídos de forma equivocada.
9. o acesso ao sistema deveria ser por meio de uma senha composta por letras e números e não apenas por números.
10. o sistema deveria permitir a exclusão de registros de pessoas que deixaram de ser clientes da instituição.
11. o sistema, após consultar os dados de um cliente, deveria permitir a impressão dos dados.
12. **Um Analista de Informática levantou os requisitos para desenvolver um sistema de gestão. Dentre os requisitos levantados,**

I. o sistema deve apresentar a tela de login e senha antes de cada transação e validar o acesso com base nas políticas de segurança organizacional.

II. o sistema deve estar disponível para a diretoria em tempo integral, ou seja, 24 x 7.

III. o tempo de resposta de uma consulta da alta administração não pode exceder a 5 milissegundos.

IV. cada Diretor que usa o sistema deve ser identificado apenas por sua matrícula de cinco dígitos seguidos do código de segurança.

V. o sistema deverá gravar um log de autenticação a cada transação completada, contendo a identificação do usuário, data e equipamento utilizado.

VI. os backups do sistema deverão ser feitos diariamente a fim de evitar a eventual perda de dados sem capacidade de recuperação.

Contêm um requisito funcional e um requisito não funcional, respectivamente, APENAS os itens

1. **II e I.**
2. V e I.
3. II e III.
4. VI e V.
5. **Requisitos não funcionais são agrupados em três grandes subgrupos. Um deles possui a seguinte definição:**

Esses requisitos especificam ou restringem o comportamento do software. Exemplos incluem os requisitos de desempenho quanto à rapidez com que o sistema deve executar e quanta memória ele requer, os requisitos de confiabilidade que estabelecem a taxa aceitável de falhas, os requisitos de proteção e os requisitos de usabilidade. (SOMMERVILLE, 2019).

A definição apresentada refere-se ao subgrupo requisitos

1. **de produto.**
2. internos.
3. externos.
4. extrapolados.
5. organizacionais.
6. **No contexto de desenvolvimento e manutenção de sistemas e aplicações, considere as seguintes afirmações sobre requisitos de software.**

I - Requisitos não funcionais são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, desempenho e tempo de resposta.

II - Requisitos não funcionais podem afetar a arquitetura de um sistema em vez de apenas componentes individuais. Por exemplo, para assegurar que sejam cumpridos os requisitos de desempenho, será necessário organizar o sistema para minimizar a comunicação entre os componentes.

III - Um único requisito não funcional, tal como um requisito de proteção, pode gerar uma série de requisitos funcionais relacionados que definam os serviços necessários no novo sistema. Além disso, também podem gerar requisitos que restrinjam requisitos existentes.

Quais afirmações estão corretas?

1. Apenas I.
2. Apenas I e II.
3. Apenas I e III.
4. Apenas II e III.
5. **I, II e III.**
6. **Considere os requisitos listados a seguir:**

I. O sistema deve ser compatível com navegadores executando em smartphones.v

II. O sistema deve permitir o cadastro de médicos, sua matrícula e suas áreas de especialização.F

III. O sistema deve permitir a emissão de relatórios contendo a alocação das salas e turmas.F

IV. O sistema deve ser desenvolvido na linguagem PHP.V

V. O sistema deve fornecer uma solução de interoperabilidade com o sistema financeiro e de recursos humanos já existentes na instituição.V

VI. O tempo de resposta a qualquer consulta deve ser de no máximo dez segundos no contexto de mil usuários simultâneos.V

São requisitos não funcionais somente os que constam em

1. I, II e IV.
2. II, III e V.
3. **I, IV, V e VI.**
4. III e VI.
5. **Qual alternativa abaixo apresenta um requisito funcional de software?**
6. A base de dados deve ser protegida para acesso apenas a usuários autorizados.
7. O tempo de resposta do sistema não deve ultrapassar 30 segundos.
8. O software deve ser operacionalizado no Sistema Operacional Windows.
9. **O software deve emitir relatórios de vendas.**
10. O tempo de desenvolvimento não deve ultrapassar três meses.

https://exerciciosresolvidos.com.br/exercicios-de-requisitos-de-software-funcionais-e-nao-funcionais

**PARTE III – Exercícios discursivos**

1. **O objetivo do gerenciamento de requisitos é assegurar a organização. Explique o que o grupo entendeu por “assegurar a organização”.**

Assegurar a organização se refere a ter mais organização nos gerenciamentos de requisitos, garantindo que o projeto seja construído de forma estruturada ao longo dele.

1. **O que são stakeholders? Relacione doze stakeholders.**

Pode ser tanto os usuários como os gerentes, funcionários, grupos e etc. É qualquer pessoa que tenha interesse direto ou indireto sobre o projeto. Pode ser considerado eles:

- Clientes

- Usuários

- Suporte

- Gerentes

- Acionistas

- Patrocinadores

- Concorrentes

- Fornecedores

- Midia

- Comunidade

- Desenvolvedores

- Colaboradores

1. **Cite três definições para requisitos.**

Funções, condições e objetivos.

1. **Qual a diferença entre requisitos funcionais e não funcionais. Cite dois exemplos de cada.**

Requisitos funcionais são os serviços ou funcionalidades que se espera de um sistema.

Exemplo:

- Fazer login com usuário e senha

- Atualizar perfil de usuário

Já os não funcionais são relacionados ao uso da sua aplicação, considerando: segurança, desempenho, usabilidade e entre outros.

Exemplo:

- Ter uma interface responsiva para diferentes telas

- Fornecer respostas rápidas pro usuário

1. **O que você entende por taxonomia de requisitos não funcionais.**

É um tipo de mapa que permite a gente observar e analisar se um requisito do sistema pertence ao grupo de não funcionais.

**PARTE IV – Desenvolvimento**

A equipe de Tecnologia da Informação (TI) irá desenvolver um sistema para automatizar o funcionamento de uma pensão em uma cidade litorânea.

O sistema a ser desenvolvido deverá possuir as seguintes funcionalidades:

1. Cadastro de reservas de quartos por telefone
   1. o cliente telefona para o hotel e solicitar a reserva ao atendente informando: acomodação, data de entrada e saída;
   2. o atendente consulta pelo nome e data de nascimento para verificar se o cliente já é cadastrado
      1. Caso seja, confirmar os dados;
      2. Caso não seja, deve-se cadastrar o cliente, que deve informar: nome, endereço, telefone, bairro, cidade, estado, data de nascimento e RG;
2. Controle de gastos dos hóspedes
3. deve-se ter o controle dos gastos de cada quarto, os gastos podem ser de telefone, diária do hotel e alimentação (valor fixo por dia);
4. Fechamento de conta
5. O sistema deve emitir uma nota com os valores gastos de telefone, diária de hotel e alimentação e o total geral incluindo 5% de serviços do hotel;

**Premissas**

* A diária do hotel varia de acordo com a acomodação, que pode ser: simples, dupla ou tripla;
* Os dados de acomodação, como: tipo, número, descrição e disponibilidade devem ser consultados no momento da reserva;
* O controle de depósito de reservas é controlado pelo sistema financeiro, que apenas informa o sistema de hotel em questão se o valor da reserva foi creditado e qual valor creditado;
* No fechamento de conta deve-se descontar o valor já pago na reserva.

**Enunciado**

Elaborar a tabela de requisitos para o sistema acima:

* 1. Requisitos funcionais (pelo menos cinco);
  2. Requisitos não-funcionais (pelo menos cinco).

Requisitos funcionais:

- Cadastrar cliente e suas informações caso não possua.

- Sistema de consulta de dias disponíveis.

- Fechar e mostrar a conta final pro usuário.

- Controle de gastos dos hospedes.

- Atualizar os status das reservas agendadas.

Requisitos não funcionais:

- Usabilidade para ser um sistema intuitivo e fácil.

- Segurança para não haver vazamentos de informações pessoais.

- Disponibilidade pra quem quiser acessar o sistema a hora que quiser.

- Desempenho para não afetar o tempo de demora nas funcionalidades do site.

- Compatibilidade para que seja disponível a todos os sistemas operacionais.