Centro Universitário Senac RS campus Pelotas

**Engenharia de Software I - Profª. Bruna Ribeiro**

**Atividade aula 2**

1. Sobre o modelo de processo de desenvolvimento de software clássico, o modelo em cascata,

realize uma pesquisa e elenque pontos fortes e também os pontos fracos sobre este modelo.

**Pontos fortes:**

* **Adequado para projetos estáveis: Funciona melhor quando os requisitos do projeto estão bem definidos e mudanças são mínimas.**
* **Fases bem definidas: O projeto é dividido em fases distintas, que ocorrem em uma sequência linear, sem sobreposições.**
* **Documentação rica: Cada fase produz saídas documentadas, que servem como base para a fase seguinte.**
* **Gerenciamento facilitado: Como as fases ocorrem de forma sequencial e linear, o gerenciamento do projeto se limita apenas a atual fase.**

**Pontos fracos:**

* **Atrasos em uma fase podem afetar todo o cronograma: Se a fase posterior depende do término da anterior, um atraso na fase de implementação, por exemplo, acarretará em um atraso em todas as demais fases posteriores (teste e manutenção), visto que estas só se iniciarão após o termino da fase atual.**
* **Identificação tardia de problemas: A falta de *feedback* contínuo (reavaliações) pode fazer com que problemas venham a ser identificados apenas em fases posteriores, fazendo com que o projeto tenha de retornar para fases já concluídas.**
* **Pouca flexibilidade para acomodar mudanças nos requisitos: Como as fases ocorrem separadamente, mudanças de requisitos fazem com que o projeto tenha de retroceder em suas fases.**

1. Leia sobre o modelo evolutivo em Espiral e cite suas principais vantagens e desvantagens do modelo.

**Vantagens:**

* **Adequado para projetos de grande escala, complexos ou com alto nível de incerteza: Os passos vão sendo repetidos (como um espiral) até que um produto seja obtido.**
* **Foco no *feedback* contínuo: Identificação contínua de riscos ao longo do processo.**
* **Abordagem progressiva: Cada ciclo inclui uma avaliação detalhada das etapas anteriores, identificando problemas e ajustando a estratégia conforme necessário, além de resultar em um incremento funcional do *software*.**
* **Adaptação a mudanças: Flexibilidade para lidar com mudanças nos requisitos, com a capacidade de ajustar o projeto em cada ciclo.**
* **Gestão de riscos eficaz: Ênfase na identificação, análise de riscos ao longo do desenvolvimento.**

**Desvantagens:**

* **Maior complexidade gerencial: Requer avaliação constante, o que pode ser mais intensivo em termos de custos.**
* **Maior tempo e custo que modelos lineares: Como existe flexibilidade para lidar com mudanças nos requisitos, o projeto pode requerer um tempo maior para ser finalizado, o que também significa custos maiores.**
* **Possível Sobrecarga de Risco: Embora a análise de riscos seja um ponto forte, se não for gerida adequadamente, pode levar a uma sobrecarga de preocupações e esforços desnecessários em áreas de menor impacto.**

1. Pesquise e descreva mais características sobre o modelo de Prototipação incluindo seus principais pontos fortes e fracos.

**Características:**

* **Iterativo e Incremental: A prototipação geralmente ocorre em ciclos iterativos, com cada iteração refinando o protótipo com base no feedback recebido, levando a melhorias graduais.**
* **Interatividade: Um protótipo é interativo e permite que os usuários experimentem as funcionalidades-chave do sistema, muitas vezes simulando as interações reais que teriam com o produto final.**
* **Rapidez: A prototipação visa criar protótipos rapidamente para permitir a validação de ideias e conceitos em estágios iniciais do projeto.**
* **Foco nas Necessidades do Usuário: Os protótipos são projetados para atender às necessidades e expectativas dos usuários finais, ajudando a refinar os requisitos e melhorar a usabilidade.**

**Pontos fortes:**

* **Validação de Requisitos: Os protótipos ajudam a identificar requisitos incompletos, ambíguos ou mal compreendidos.**
* **Melhoria da Comunicação: Protótipos visuais facilitam a comunicação entre desenvolvedores, designers e stakeholders.**
* **Usabilidade Aprimorada: Os protótipos permitem testar a usabilidade do sistema antes da implementação.**

**Pontos fracos:**

* **Possível Desvio de Escopo: A criação de protótipos pode levar a um aumento não planejado no escopo do projeto.**
* **Investimento de Tempo: A criação de protótipos pode consumir recursos e tempo que poderiam ser alocados para o desenvolvimento real.**

1. Ao final da sua pesquisa, elabore uma tabela contendo os modelos de processos de desenvolvimento de software estudados, suas etapas, sua descrição, pontos fortes de cada uma e pontos fracos, e encontre exemplos ou tipos de softwares que possam ser adequados ao modelo. E encontre na sua pesquisa mais algum modelo, complemente a tabela abaixo com os mesmos campos!

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modelos** | **Etapas** | **Descrição** | **Pontos Fortes** | **Pontos Fracos** | **Exemplos de Software que se adequariam ao modelos** |
| Modelo em  Cascata | * **Levantamento de Requisitos** * **Análise** * **Projeto** * **Implementação** * **Testes** * **Manutenção** | **Um dos modelos de processo de desenvolvimento de *software* mais antigos e tradicionais, é caracterizado por uma abordagem sequencial e linear, onde o desenvolvimento do *software* ocorre em fases distintas, com cada fase sendo concluída antes que a próxima comece.** | * **Adequado para projetos estáveis** * **Fases bem definidas** * **Documentação rica** * **Gerenciamento facilitado** | * **Atrasos em uma fase podem afetar todo o cronograma** * **Identificação tardia de problemas** * **Pouca flexibilidade para acomodar mudanças nos requisitos** | * **Sistemas de Processos Industriais** * **Sistemas de Controle de Tráfego Aéreo** |
| Modelo em Espiral | * **Planejamento** * **Análise de Riscos** * **Engenharia** * **Avaliação** | **Combina elementos de abordagens sequenciais e iterativas. Ele foi proposto por Barry Boehm em 1986 como uma maneira de lidar com a complexidade e os riscos inerentes ao desenvolvimento de *software*.** | * **Adequado para projetos de grande escala, complexos ou com alto nível de incerteza** * **Foco no feedback contínuo** * **Abordagem progressiva** * **Adaptação a mudanças** * **Gestão de riscos eficaz** | * **Maior complexidade gerencial** * **Maior tempo e custo que modelos lineares** * **Possível Sobrecarga de Risco** | * **Sistemas de militares de defesa** * **Sistemas de TI.** |
| Prototipação | * **Identificação dos Requisitos** * **Escolha do Tipo de Prototipação** * **Desenvolvimento do Protótipo Inicial** * **Avaliação e Feedback** * **Revisão** * **Implementação** | **Essencial no processo de desenvolvimento de *software*, envolve a criação de versões iniciais e simplificadas de um sistema para validar conceitos, testar funcionalidades e obter feedback dos usuários antes da implementação completa.** | * **Validação de Requisitos** * **Melhoria da Comunicação** * **Usabilidade Aprimorada** | * **Possível Desvio de Escopo** * **Investimento de Tempo** | * **Aplicações Web (UI/UX).** * **Aplicativos.** |
| **Scrum** | * **Dividido em *sprints* (períodos de tempo fixos, geralmente de 2 a 4 semanas)** * **Identificação dos requisitos** * **Planejamento** * **Implementação** * ***Daily Scrum* (reunião diária).** * **Revisão.** * **Retrospectiva.** * ***Deployment*.** | **Enfatiza a colaboração, a transparência e a responsabilidade, com papéis definidos como *Product Owner*, *Scrum Master* e Equipe de Desenvolvimento.** | * **Adaptabilidade e Flexibilidade** * **Entrega Incremental e com Foco na Melhoria Contínua** * **Colaboração e Comunicação** * **Envolvimento do Cliente** | * **Implementação Complexa** * **Priorização de Requisitos Difícil** * **Potencial Sobrecarga da Equipe** * **Dependência de Experiência e Maturidade da Equipe** | * ***eCommerces*/*Marketplaces*** * **Aplicativos.** * **Redes sociais.** |