03/02/2023

Introdução à Eng de Software

* História:
  + O conceito foi proposto em 1968 após a crise do software.
  + Os sistemas não eram confiáveis, entregue fora do prazo e cheio de bugs.
  + Durante 1970 e 1980 foi desenvolvida uma série de técnicas e métodos afim de resolver o problema anterior.
  + A velocidade do hardware foi e está sendo muito superior a velocidade da evolução do software, devido a necessidade da força do homem.
  + As organizações estão cada vez mais dependentes dos softwares
* Problemas:
  + Processo:
    - Cronogramas muito otimistas
    - Gerencia de risco inexistente
    - Falha de contratação de recursos
    - Planejamento insuficiente
    - Abandono do planejamento por problemas de tempo
    - Gasto de tempo durante a concepção
    - Analise ou arquitetura
    - Design inadequado
    - SQA ou CM inexistente 🡪 qualidade do software
  + Produto:
    - Requisitos mal formulados
    - Desenvolvedores com tecnologias folheadas em ouro
    - Negociação Puxa-empurra
  + Tecnologia:
    - Superestimar ganhos no uso de uma nova tecnologia
    - Trocar de ferramentas no meio do projeto
    - Falha de controle do código
* Software e suas Aplicações:
  + Software:
    - Conjunto de instruções que quando executadas fornecem características, funções e desempenho **DESEJADOS**
    - Estruturas de dados que possibilitam aos programas manipular informações adequadamente
    - informação descritiva, tanto na forma impressa como na virtual, descrevendo a operação e o uso dos programas.
  + Características:
    - Não se desgasta
    - Construídos sob encomenda
    - Toda mudança aumenta a falha
  + Diagrama

    Descrição gerada automaticamenteAplicações:
    - De sistemas
    - De tempo real
    - De aplicação
    - Científico e de engenharia
    - Embutido
    - IA
    - Livre
    - Computação Ubíqua
  + Parou no slide 37
* Engenharia de Software:
  + O estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de **maneira econômica**, que seja **confiável** e funciona de forma **eficiente em máquinas reais**.

Aula 10/02

Processos de Software:

* Processo:
  + Sequência de passo realizados com um propósito.
  + Integra pessoas, ferramentas e procedimentos.
  + É o que as pessoas fazem, utilizando procedimentos, métodos, ferramentas e equipamentos para transformar inputs em outputs de valor para seus clientes
  + Matéria-prima para software: Requisito de software ( o que ele faz)
  + Processo de Desenvolvimento de Software 🡪 pode ser definido como um conjunto de atividades, métodos, praticas e transformações que pessoas utilizam para desenvolver ou dar manutenção em software ou em seus produtos associados
  + Processo de software é parecido com Engenharia de software , porém o segundo tbm engloba tecnologias que fazem parte do processo
* Modelo de Processo Genérico:
  + Atividades metodológicas básicas – planejamento de como você resolverá o problema:
    - Comunicação
    - Planejamento
    - Modelagem
    - Construção 🡪 se começarmos daqui, vai dar bosta
    - Entrega
  + Atividades de apoio:
    - Controle e acompanhamento do projeto
    - Administração de riscos
    - Garantia de qualidade
    - Revisões técnicas
    - Medição
    - Gerenciamento de configuração de software
    - Gerenciamento do reuso
    - Preparo e produção de artefatos
  + Fluxo de processos:
    - Linear
    - Iterativo – pode voltar
    - Evolucionário – pode voltar e evolui, usado na maioria dos casos
    - Diagrama

      Descrição gerada automaticamenteParalelo – atividades feita em paralela
  + Modelo:
    - Atividades do processo são definidas de acordo com a complexidade no projeto, ou seja, projetos diferentes demandam conjuntos de tarefas diferentes. A equipe de software escolhe o conjunto de tarefas fundamentada no problema e nas características do projeto.
  + Padrões de Processo:
    - Descreve um problema relacionado ao processo que pode ser encontrado em um trabalho de engenharia de software
    - Identifica o ambiente onde problema é normalmente encontrado
    - Sugere soluções para o problema
    - Tipos:
      * De estágio 🡪 define um problema associado a uma atividade metodológica
      * De tarefa 🡪 define um problema associado a uma ação e sua prática
      * De fases 🡪 sequência de atividades metodológicas
  + Avaliação e Aperfeiçoamento de Processos:
    - SCAMPI
    - CBA IPI
    - SPICE
    - ISO
  + Modelo de Processo Prescritivo:
    - Esses modelos defendem uma abordagem ordenada
    - Tentam trazer ordem ao caos(code and fix)
    - Prescrevem um conjunto de elementos do processo
    - Modelos tradicionais
    - Exemplos:
      * Cascata/ciclo de vida clássico
      * Incremental 🡪 combina elementos do linear e paralelo , cada sequencia linear produz um incremento executável do software(núcleo)
      * Evolucionário 🡪 interativos , apresentam características que possibilitam desenvolver versões cada vez mais completas do software

Aula dia 16/02/2023

* Modelo Cascata - problemas:
  + Utópico, ou seja , projetos reais não utilizam ele
  + Difícil estabelecer todas as necessidades do cliente no inicio do projeto
  + Erros graves podem não ser detectados até o programa operacional ser revisto
  + Cliente deve ser paciente , pois uma versão do sistema não estrá disponível antes de estarmos próximos ao fim
* Modelo de processo incremental: 🡪 cada sequencia linear produz um incremento executável do software 🡪**produto essencial**
* Modelos de processos evolucionário:
  + Prototipação e Modelo Espiral
  + Iterativos ou incrementais
  + Problema:
    - Devido a burocracia das grandes operações pode gerar uma incompatibilidade entre o modelo, por ser mais informal
* Modelos de processos Especializado:
  + Desenvolvimento baseado em componentes 🡪 importância do reuso de software
  + Modelo de métodos formais 🡪 ênfase na **especificação matemática** dos requisitos de software
* Processo de software Pessoal (PSP):
  + Importância de aprender com os erros
  + O PSP (Personal Software Process) possui cinco atividades estruturais:
    - Planejamento
    - Projeto de alto nível
    - Revisão de projeto de alto nível
    - Desenvolvimento
    - Autópsia (análise postmortem)
* Processo de software em Equipe:
  + Baseado em lições aprendidas no anterior
  + Objetivo é criar equipes **autodirigidas** para produzir software de alta qualidade
  + O TSP (Team Software Process) possui cinco atividades metodológicas:
    - Lançamento do projeto
    - Projeto de alto nível
    - Implementação
    - Integração e testes
    - Autópsia (análise postmortem)
* Processo de manutenção de sistemas:
  + Identificar e corrigir o problema
  + Analisar impactos
  + Atualizar documentação
* Processos e Equipes no Modelo Cascata(Modelo monolítico – Analista , Projetista e Desenvolvedor)
  + Planejamento
  + Não é escalável
  + Requisitos viram desenhos, e logo depois códigos
  + Antítese do modelo iterativo e incremental
* Processos e Equipes no Modelo Iterativo(Modelo Hierárquico)
  + Arquiteto de Sistema com visão macro
  + Analista de Sistemas com visão macro
  + Desenvolvedores com visão micro
* Desafios dos processos de software:
  + Inteligência compartilhada
  + Integração
  + Comunicação
  + Dividir conquistas
  + Visão integrada
  + Ciclo de vida
  + Gerenciamento, controle e resultado
* Processo Unificado:
  + Baseado na construção de software a partir de componentes interconectados através de interfaces bem definidas
  + Chave:
    - Caso de uso
    - Centrado na Arquitetura
    - Iterativo e Incremental

Aula dia 24/02

Desenvolvimento Ágil de Software

* Agilidade:
  + Não significa ser rápido e sim fazer direito com o objetivo de não desperdiçar tempo
  + Entregar o mais rápido possível BEM FEITO
* Processo Ágil:
  + Cenários construídos pelos clientes com seus requisitos
  + Planejamento tem vida curta
  + Entrega de múltiplos incrementos de software , pode ocorrer mudanças
* Extreme Programming (XP)
  + Proposto por Kent Beck em 2004
  + Valores:
    - Comunicação 🡪 Relação estreita e informal
    - Simplicidade 🡪 Fazer somente o necessário
    - Feedback 🡪 devem ser constantes , á medida do desenvolvimento , com testes e entregas
    - Coragem 🡪 Pensar no presente , mas bem feito
    - Diagrama

      Descrição gerada automaticamenteRespeito
  + Planejamento:
    - HU + custo + data
  + Desenho:
    - KISS (keep it simple)
  + Codificação
    - Construção de teste antes da codificao
    - Programação em a pares
  + Teste
    - Executados diariamente
* Desenvolvimento de software adaptativo:
  + Caracteristicas:

Planejamento dirigido a missão

* + - Foco no desenvolvimento baseado em componentes
    - Uso de “time-boxing”
    - Consideração explícita de riscos
    - Ênfase na colaboração para entendimento dos requisitos
    - Ênfase na “aprendizagem” durante o processo
* Scrum:
  + Diagrama

    Descrição gerada automaticamenteCom funciona?
  + Papeis fundamentais:
    - Product Ownder
    - Scrum Master
    - Scrum team
  + Um projeto Scrum é iniciado quando o Product Owner (PO), responsável por obter informações dos stakeholders, ou usuários que os representem, elaboram uma lista de requisitos e criam um **Backlog de Produto**(list de requisitos priorizada).
  + Criação de BP 🡪 por meio de histórias curtas de usuários , que viram planejamento de releases e sprints
  + Criação de Sprint 🡪 identificar tarefas pelas HU + estivatimava de tempo + Task board
  + Reunioes de sprints e diárias de 15 min
* Quando não usar métodos ou processos ágeis?
  + Condições do mercado são estáveis e previsíveis
  + Requisitos são claros no início do projeto e irão permanecer estáveis
  + Clientes não estão disponíveis para colaboração frequente
  + Sistema semelhante já foi feito antes; logo, já se conhece a solução
  + Problemas podem ser resolvidos sequencialmente, em silos funcionais
  + Clientes não conseguem testar partes do produto, antes de completo
  + Mudanças no final do projeto são caras ou impossíveis
  + Impacto de mudanças provisórias pode ser catastrófico

Aula 24/12/2023

Histórias de Usuários

* O que é?
  + “Uma história de usuário é uma descrição curta e simples de um requisito contada a partir da perspectiva da pessoa que deseja a nova funcionalidade do sistema”
  + Modelo : 
* Historia tem foco no valor definido pelo usuário , requisito tem foco no negócio
* Características de boas historias:
  + Independentes
  + Abertas para negociação
  + Agregar valor
  + Estimáveis
  + Sucintas
  + Pequena
  + Testaveis