ENGENHARIA DE SOFTWARE

INTRODUÇÃO À ENGENHARIA DE SOFTWARE - UNIDADE 01

Abrange Processos e Conjuntos de Métodos e Ferramentas que Permitem um Desenvolvimento de Qualidade e Dentro dos Prazos desejados.

O SOFTWARE

Se tornou algo Indispensável em nossas vidas por Fazer a Informação Transitar pelo mundo todo. Consistem em Programas que podem ser Executados e seus Dados além de toda sua Documentação.

METODOLOGIAS TRADICIONAIS

MODELO CASCATA

O Modelo Cascata é Usado para Desenvolver Programas e descreve por meio de Etapas ou Fases seu Ciclo de desenvolvimento. Uma Fase do processo Depende do Artefato Gerado pela Fase Anterior. Em caso de Falhas ocorre o Retorno à Fase Anterior para Sanar Problemas que caso fossem Adiante causariam Danos ao Processo.

ETAPAS DO MODELO CASCATA

REQUISITOS (LEVANTAMENTO DO QUE O SISTEMA DEVERÁ FAZER)
PROJETO (PENSAR SOBRE COMO O SISTEMA DEVERÁ FAZER)
IMPLEMENTAÇÃO (DESENVOLVIMENTO DO PRODUTO EXECUTÁVEL)
TESTES (BUSCA COM O OBJETIVO DE REVELAR DEFEITOS)
MANUTENÇÃO (CORREÇÕES EM PRODUTOS JÁ ENTREGUES)

METODOLOGIAS ÁGEIS

EXTREME PROGRAMMING (XP)

O Extreme Programming é uma Metodologia Adequada para projetos cujos Requisitos se Alterem frequentemente. Também é Recomendada para Ocasiões onda Existe a Busca por Partes Executáveis no Início.

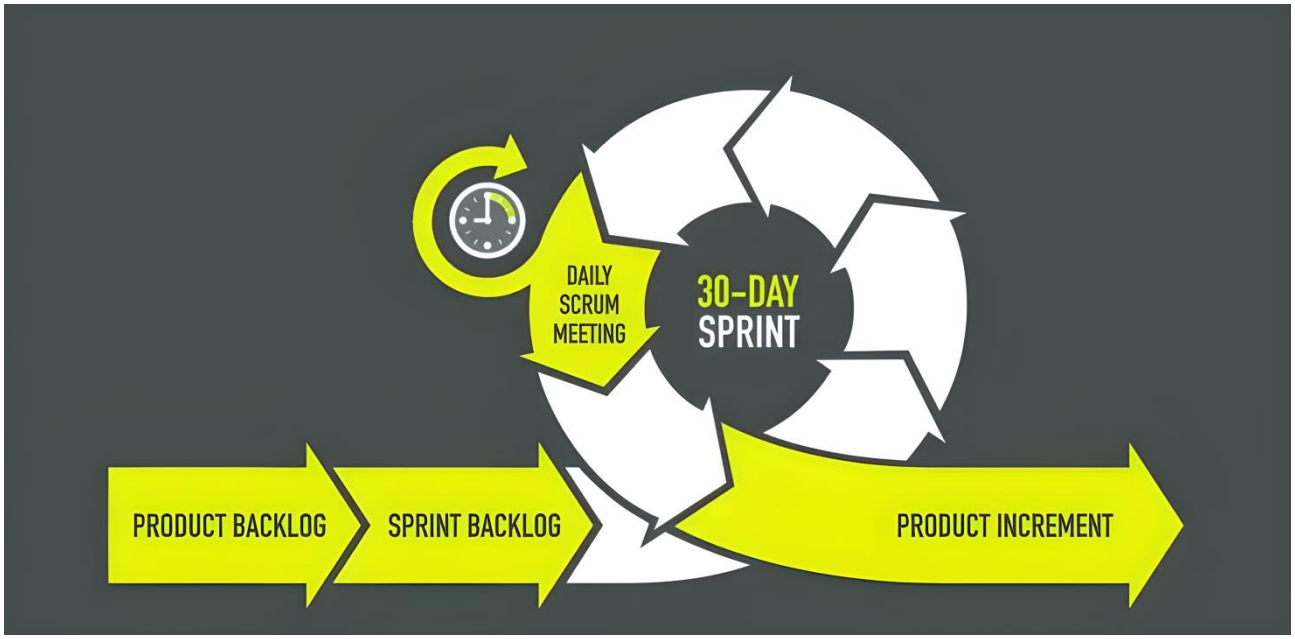
PROCESSOS DO EXTREME PROGRAMMING

PLANEJAMENTO (LEVANTAMENTO DE REQUISITOS) (ESTÓRIAS)
PROJETO (GUIA DE IMPLEMENTAÇÃO DE CADA ESTÓRIA)
CODIFICAÇÃO (TESTES PARA CADA UMA DAS ESTÓRIAS)
TESTES (TESTES PODEM SER USADOS A QUALQUER MOMENTO)

PILARES DO EXTREME PROGRAMMING

FEEDBACK (TROCA DE INFORMAÇÕES ENTRE CLIENTE E EQUIPE)
COMUNICAÇÃO (EVITAR O TRABALHO ESPECULATIVO)
SIMPLICIDADE (DESENVOLVER APENAS O SUFICIENTE)
CORAGEM (LEVAR ADIANTE AS PRÁTICAS DA METODOLOGIA)

SCRUM



PRODUCT BACKLOG (LISTA DE FUNCIONALIDADES DESEJADAS)
SPRINT BACKLOG (TAREFAS SELECIONADAS DO PRODUCT BACKLOG)
SPRINT (PROCESSO DE EFETIVA CONSTRUÇÃO DO SOFTWARE)
SCRUM MEETING (REUNIÕES REGULARES DOS DESENVOLVEDORES)
PRODUCT INCREMENT (FUNCIONALIDAE DESENVOLVIDA NA SPRINT)

MEMBROS SCRUM

SCRUM MASTER

Agente com Amplo Conhecimento do Modelo que Preza pelo Correto Uso do Mesmo Durante Todas as Etapas do Projeto.

PRODUCT OWNER

Agente Responsável pelo Projeto. Indica os Requisitos mais Importantes a serem Tratados nas Sprints.

GERENCIAMENTO DE CONFIGURAÇÃO DO SOFTWARE

O GCS Consiste em um Conjunto de Práticas que Controlam e Notificam Correções e Adaptações. Tem como Objetivo Assegurar um Processo de Desenvolvimento Organizado e Rastreável.

CONTROLE DE VERSÕES

O Controle de Versões consiste no Meio pelo qual o GCS Controla de forma Consistente as Modificações Realizadas em um Sistema.

REPOSITÓRIOS

Local onde Programas em Desenvolvimento e os Demais Arquivos são Armazenados e podem ser Acessados de forma Controlada por Todos os Envolvidos no Desenvolvimento do Produto.

BASELINES

Representam Conjuntos de Itens de Configuração que servem de Base para as Etapas Seguintes de Desenvolvimento.

BRANCHES

Implemento de Novas Funcionalidades por uma Equipe que é Realizada em Paralelo porém de Forma Isolada de Outros Desenvolvedores.

FERRAMENTAS DE CONTROLE DE VERSÃO

GIT (FERRAMENTA PARA CONTROLE DE VERSÕES)
GITHUB (EQUIVALENTE AO REPOSITÓRIO DO GIT)

INTRODUÇÃO À QUALIDADE DE SOFTWARE - UNIDADE 02

A Qualidade tem como Objetivo Atender as Necessidades e Expectativas do Cliente além de Cumprir com os Requisitos que foram Acordados.

REQUISITOS FUNCIONAIS

Funções que um Sistema ou algum de seus Componentes deve Realizar.

REQUISITOS NÃO FUNCIONAIS

Relacionado aos Requisitos Técnicos que o Sistema deve Desempenhar.

VANTAGENS DA GARANTIA DA QUALIDADE

Os Processos de Garantia da Qualidade Dependem de um Esforço Coletivo que Proporciona a Economia de Recursos. As Vantagens Obtidas com o Uso desses Recursos de Qualidade nas diversas atividades de desenvolvimento Demoram a serem Sentidas no Cotidiano. Isso Ocorre Conforme o Tempo e o Constante Uso dessas Ferramentas.

QUALIDADE DE PRODUTO

Tem como Objetivo Central Padronizar as Atividades e a Forma com que se Avalia a Qualidade de um Produto.

ISO 9126

FUNCIONALIDADE
- **ADEQUAÇÃO**
- **ACURÁCIA**
- **INTEROPERABILIDADE**
- **CONFORMIDADE**
- **SEGURANÇA DE ACESSO**

CONFIABILIDADE
- **MATURIDADE**
- **TOLERÂNCIA A FALHAS**
- **RECUPERABILIDADE**

USABILIDADE
- **INTELIGIBILIDADE**
- **APREENSIBILIDADE**
- **ATRATIVIDADE**

EFICIÊNCIA
- **TEMPO**
- **RECURSOS**

MANUTENIBILIDADE
- **MODIFICABILIDADE**
- **ESTABILIDADE**
- **ESCALABILIDADE**

PORTABILIDADE
- **ADAPTABILIDADE**
- **ANALISABILIDADE**
- **INTEROPERABILIDADE**

QUALIDADE DE PROCESSO

Ferramentas de Qualidade de Processos tem a Intenção de Sanar Erros e Padronizar Atividades. Mas Antes se faz Necessário Mapear os Processos para Identificar a Ordem que as Atividades são Executadas.

NÍVEIS DE DETALHAMENTO DOS PROCESSOS

DESCRITIVO (DESCRIÇÃO BÁSICA E ABRANGENTE DOS PROCESSOS)
ANALÍTICO (DETALHAMENTO ATIVIDADES DE DESENVOLVIMENTO)
EXECUTÁVEL (DETALHAMENTO DE FUNCIONLAIDADES E SERVIÇOS)

MODELOS DE MATURIDADE

Modelos de Maturidade são Ferramentas que Permitem o Conhecimento Profundo dos Processos e das Demais Partes que Envolvem um Projeto.

CMM

NÍVEIS DE MATURIDADE

INICIAL (INEXISTE CONTRLE DE PROCESSOS)
REPETITIVO (CONTROLES DE PROCESSO BÁSICOS SÃO USADOS)
DEFINIDO (BOAS PRATICAS SÃO ESTABELECIDAS COMO PADRÃO)
GERENCIADO (USO FERRAMENTAS ESTATISTICAS)
OTIMIZADO (POSSIBILIDADE DE REPENSAR ALGUNS PROCESSOS)

CMMI

NÍVEIS DE MATURIDADE

INCOMPLETO (PROCESSOS NÃO EXECUTADOS EM SUA TOTALIDADE)
EXECUTADO (PROCESSOS DEFINIDOS)
GERENCIADO (PLANEJAMENTO E MONITORAMENTO DE PROCESSOS)
DEFINIDO (PROCESSO SERVE COMO PADRÃO)
GERENCIADO (ÁNALISES ESTATISTICAS DOS PROCESSOS)
OTIMIZADO (FOCO NA MELHORIA CONTÍNUA)

NORMAS ISO DE QUALIDADE DE PROCESSOS (IS0 9001)

Sistema de Qualidade que visa Garantir a Otimização dos Processos. Essa Norma Também é Conhecida como SQA por Algumas Pessoas.

MELHORIA DE PROCESSOS INDIVIDUAIS E DE EQUIPE (PSP)

Essa Ferramenta Visa às Pessoas que Desenvolvem os Sistemas e tem como Objetivo promover o Desenvolvimento com Enfoque na Habilidade Individual dos Colaboradores. Segundo essa Ferramenta para Melhoria das Habilidades Individuais se Deve Observar os Erros e cometidos Para que possam Ser Corrigidos e Aprendidos pelo Desenvolvedor.

CONCEITOS DE TESTES DE SOFTWARE - UNIDAE 03

Mecanismos onde Produtos passam por Processos que Atestem sua Propensão para Executar Adequadamente suas Funções com Elevados Níveis de Qualidade. Tais Mecanismos são Conhecidos como Testes.

TESTES DE SOFTWARE

Testar Depende de um Conjunto de Ações e Procedimentos Executados por Diversos Elementos de uma Equipe. Consistem em uma Sequência de Ações que tem como Objetivo Encontrar Problemas e Aumentar a Percepção sobre a Qualidade Geral de um Sistema.

PLANO DE TESTES

Testes Precisam ser Planejados com Antecedência e Executados com Base em um Modelo Padrão. Destinar a Atividade de Teste ao Mesmo Time que Desenvolveu o Produto Certamente Não é Recomendado pois é Grande a Chance da Equipe Entender que deve Proteger Seu Programa. Outra Razão para Evitar a Designação dos Criadores do Programa como seus Testadores é o fato de que um Terceiro Poderá Detectar Falhas no entendimento de Requisitos que Passaram Despercebidas.

CASOS DE TESTE

Par Formado por uma Entrada dada no Programa e sua Correspondente Saída. Devemos Entender os conceitos de Entrada como o Conjunto de Dados Necessários Para uma Execução e Saída como seu Resultado.

ETAPAS DE TESTE

PLANEJAMENTO (DEFINIR OS REQUISITOS DO TESTE)
PROJETO DE CASOS DE TESTE (DEFINIR OS CASOS DE TESTE)
EXECUTAR O PROGRAMA COM OS CASOS DE TESTE (REALIZAR TESTE)
ANÁLISE DOS RESULTADOS (VERIFICAR OS RESULTADOS)

RESULTADOS

PASSOU (TODOS OS PASSOS FORAM EXECUTADOS COM SUCESSO)
FALHOU (NEM TODOS OS PASSOS OBTIVERAM SUCESSO)
BLOQUEADO (O TESTE NÃO PÔDE SER EXECUTADO)

DEPURAÇÃO

Ocorre como Consequência de um Teste que Descobre uma Falha. Saber que Existe um Problema Causador de Erro no Programa é Diferente de Saber em Qual ou Quais Linhas o Problema está Localizado. Portanto Depurar Consiste no Processo de Localizar esses Defeitos no Código.

TÉCNICA DE TESTE FUNCIONAL E TESTE DE FUNCIONALIDADE

Essa Abordagem tem uma Visão Externa do Produto. O Teste é Realizado nas Funções do Programa. Tem o Objetivo de Observar se o Algoritmo Produz os Resultados Esperados Conforme as Entradas dadas. O Testador Desconhece os Detalhes Internos do Sistema e Baseia seu Julgamento apenas nos Resultados Obtidos. Os Testes de Funcionalidades Priorizam Interações com o Usuário e a Navegação no Sistema.

TÉCNICA DE TESTE ESTRUTURAL

Conhecidos por Serem Baseados na Arquitetura Interna do Programa.

TESTES DE APLICAÇÕES WEB E MÓVEIS

Atividades organizadas de Teste Executadas com o Objetivo de Descobrir Erros em Aspectos Fundamentais de Funcionamento. Esses Testes são Realizados tanto pelos Profissionais Técnicos e Usuários como também pelos Clientes e Gerentes do Projeto.

TESTES DE APLICAÇÕES ORIENTADAS A OBJETOS

Pelas Características Próprias dos Programas Orientados a Objetos os Testes Devem Considerar a Presença de Subsistemas em Camadas que Encapsulam Outras Classes. Se torna Preciso Testar em Níveis Diferentes para que Erros eventualmente Ocorridos Durante as Interações entre as Classes sejam Descobertos e Sanados.

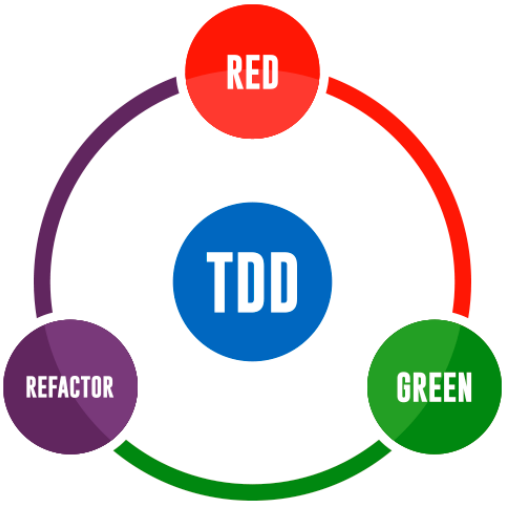
TESTE DE UNIDADE (TESTAR CADA MÉTODO DE FORMA ISOLADA)
TESTE DE MÓDULO (TESTAR UM CONJUNTO DE UNIDADES)
- INTERMÉTODO (TESTAR MÉTODOS PÚBLICOS DE UMA CLASSE)
- INTRACLASSE (TESTAR INTERAÇÕES ENTRE MÉTODOS PÚBLICOS)
- INTERCLASSE (TESTAR INTERAÇÕES DE CLASSES DIFERENTES)

TESTE DE SISTEMA (TESTE DO SISTEMA COMPLETO)

DESENVOLVIMENTO ORIENTADO A TESTES (TDD)

O Momento da Aplicação dos Testes é Tradicionalmente um dos Últimos. Conforme o Passar do Tempo essa Prática Experimentou uma Evolução Impulsionada pelo Aprimoramento das Metodologias Ágeis que tornou Possível a Aplicação dos Testes em Outros Momentos. Essa Evolução Recebeu o Nome de Desenvolvimento Orientado a Testes e tem como Objetivo Fazer com que o Desenvolvedor Escreva Testes Automatizados constantemente Durante o Processo de Desenvolvimento e Antes mesmo de Implementar o Código. Essa Inversão Compele o Desenvolvedor a Escrever um Código de Melhor Qualidade.

CLICO DO DESENVOLVIMENTO ORIENTADO A TESTES



VERMELHO (ESCREVER TESTE QUE FALHA INTENCIONALMENTE)
VERDE (ESCREVER CÓDIGO QUE PERMITE QUE O TESTE PASSE)
REFATORAR (MELHORAR ESTRUTURAÇÃO DO CÓDIGO)

FERRAMENTA PARA TESTES AUTOMATIZADOS DE SOFTWARE

SELENIUM

SELENIUM WEBDRIVER (SISTEMAS BASEADOS EM NAVEGADOR)
SELENIUM IDE (GRAVAR E REPRODUZIR OS TESTES) (EXTENSÃO)
SELENIUM GRID (TESTES EM VÁRIAS MÁQUINAS AO MESMO TEMPO)

FUNDAMENTOS DE AUDITORIA DE SISTEMAS - UNIDADE 04

As Atividades de Auditoria tem como principal Objetivo Analisar Parcial ou Globalmente os Processos e Sugerir Ações Corretivas ou Melhorias.

PROCESSOS (ANÁLISE DAS ATIVIDADES DE DESENVOLVIMENTO)
DESENVOLVIMENTO (ANÁLIASE DOS ERROS DE PROGRAMAÇÃO)
TESTES (ANÁLISE DA EFICÁCIA DOS TESTES EFETUADOS)
SEGURANÇA (TRATATIVAS QUANTO À PROTEÇÃO DOS DADOS)
ESTRUTURA (VOLTADA ÀS ATIVIDADES ADMINISTRATIVAS)

O AUDITOR

Agente com Amplo Conhecimento para Conduzir Atividades Relacionadas ao Desenvolvimento de Auditorias.

CICLO DE VIDA DA AUDITORIA

CRONOGRAMA (DEFINIR DATAS DA AUDITORIA)
PLANEJAMENTO (DEFINIR QUEM E O QUE DEVE SER AUDITADO)
AUDITORIA (REALIZAR A AUDITORIA)
REPORTE (CONHECER OS RESULTADOS)

CONTROLES GERAIS DE AUDITORIA DE SISTEMAS

Estruturas Internas das Empresas e suas Políticas Administrativas além dos Procedimentos Usados nas Atividades como um Todo. Os Controles Gerais são Operacionalizados da Portaria à Direção da Empresa. Assim os Colaboradores Criam o Ambiente Corporativo. Durante um Processo de Auditoria em que se Tem como Objetivo a Avaliação de um Sistema é Necessário Compreender Como esse Controle Age sobre esse Sistema.

CATEGORIAS DA ANÁLISE DOS CONTROLES GERAIS

CONTROLE ORGANIZACIONAL (POLITICAS INTERNAS DA EMPRESA)
CONTROLE GERAL DE SEGURANÇA (GERENCIAMENTO DE RISCOS)
CONTINUIDADE DE SERVIÇO (TRATATIVAS DE INCIDENTES)
CONTROLE DE SOFTWARE (CONTROLAR DE ACESSOS AO SISTEMA)
CONTROLE DE ACESSO (DETECTAR ACESSOS INAUTORIZADOS)

SOFTWARES DO SISTEMA

Conjunto de Programas Desenvolvidos para Gerenciar e Controlar as Atividades de Processamento de Dados.

EXEMPLOS

SISTEMA OPERACIONAL
SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS
SISTEMA DE BIBLIOTECAS

CONTROLE DE APLICATIVOS

O Controle de Aplicativos pode ser Definido Como as Funcionalidades que são Executadas pelos Sistemas os quais possuem Funções de Entrada e Processamento e Saída. Se Deve Atestar que em Todas essas Etapas Ocorra a Garantia da Integridade dos Dados.

CONTROLE DE ENTRADA DE DADOS (SCRIPTS)

Controles de Entrada são Desenvolvidos para Garantir que os Dados sejam Inseridos no Sistema de Forma Correta.

CONTROLE DE PROCESSAMENTO DE DADOS (LOGS)

Controles de Processamento devem Garantir que os Dados de Entrada Executados dentro do Sistema gerem Saídas Coerentes.

CONTROLE DE SAÍDA DE DADOS (RELATÓRIOS)

Controles de Saída de dados são Definidos como Ferramentas de Garantia da Integridade de Forma Consistente.

MANUTENÇÃO E EVOLUÇÃO DE SOFTWARE

Sistemas são Sequências Lógicas de Algoritmos cujo Intuito é Atender os Objetivos estabelecidos. Tais Objetivos estão Suscetíveis a Mudanças de Requisitos e Ambiente. Esse Processo de Envelhecimento é Inevitável e Exige algumas Mudanças para Garantir a Continuidade do Sistema.

TIPOS DE ENVELHECIMENTO DE SOFTWARE

FALHA DE ADEQUAÇÃO (ERROS NA ADEQUAÇÃO DOS REQUISITOS)

(OCASIONA PERDA DA INTEGRIDADE E DA CONFIABILIDADE)
FALHA NA MUDANÇA (ATUALIZAÇÕES OU IMPLEMENTAÇÕES)
(IMPACTAM NEGATIVAMENTE FUNCIONALIDADES EXISTENTES)

TIPOS DE ATIVIDADES DE MANUTENÇÃO DE SOFTWARES

O Conhecimento da Classificação dos Tipos de Manutenção Permite que tanto um Desenvolvedor quanto um Gestor se Posicione quanto às Reais Necessidades do Sistema Tornando o Direcionamento de Recursos Menos Complicado. Existem Diferentes Motivos para Evoluir um Sistema.

ADAPTATIVA (ATENDER NOVOS REQUISITOS)
CORRETIVA (CORRIGIR FALHAS OU OUTRAS INCONFORMIDADES)
EVOLUTIVA (INSERIR NOVAS FUNCIONALIDADES)

TÉCNICAS E FERRAMENTAS PARA MANUTENÇÃO DE SOFTWARES

ESTRUTURA DO CÓDIGO

Um Determinado Código de Programação deve se Legível e Intuitivo. As Técnicas de Indentação e Comentários Descrevendo Partes Importantes do Código Também devem estar Presentes.

VERSIONAMENTO

Apontar as Modificações e as Atualizações Realizadas em um Sistema Por Meio de Numeração ou Comentários é uma Boa Pratica a ser Adotada.

REENGENHARIA DE SOFTWARE

O Processo de Reengenharia é uma Forma de Reorganizar o Sistema com a Finalidade de se Apresentar um Desempenho mais Aceitável. Alguns Fatores como Falta de Evolução ou Excesso de Contínuas Mudanças Promovidas por Equipes Diferentes acabam Degradando os Serviços do Sistema. Fazer uma Reconstrução com a Correção de Erros e Falhas além de Adequar as Evoluções Necessárias é uma Solução. A Reengenharia tem o Objetivo de Reimplementar Sistemas.

MELHORAR A MANUTENÇÃO
REDOCUMENTAR O SOFTWARE
REESTRUTURAR O SISTEMA

ENGENHARIA REVERSA

Métodos de Engenharia Reversa Tratam de Reestruturar Códigos de forma a Obter uma Melhor Compreensão das Funcionalidades.