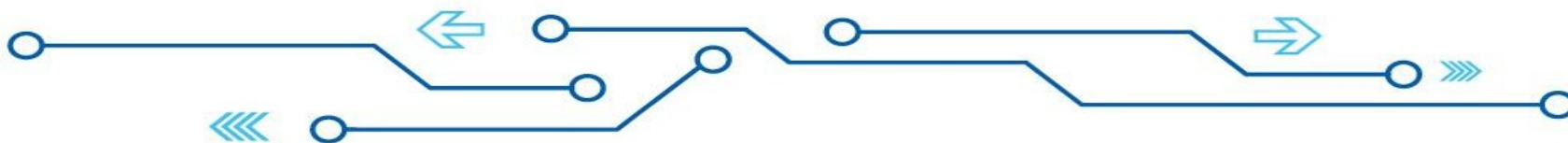


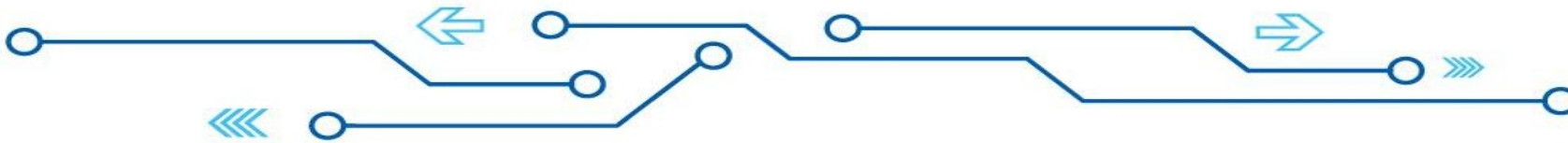


- Engenharia de Software
- Prof Elisabete



Roteiro

- O que é SWEBOK;
- Objetivos do SWEBOK;
- Áreas de Conhecimento do SWEBOK.



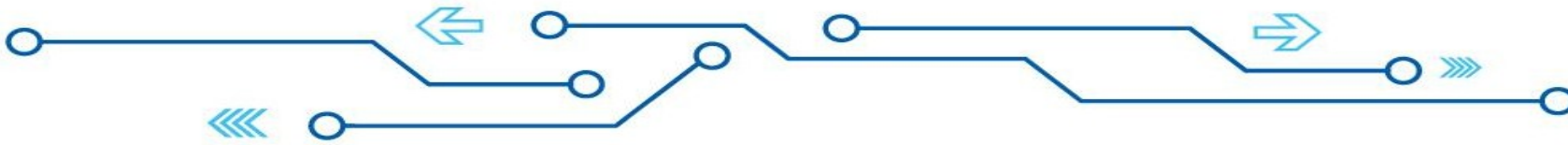
O que é SWEBOK

- É um guia para o Corpo de Conhecimento de Engenharia de Software, ou seja, fornece a definição do corpo de conhecimento aceito e práticas recomendadas;
- É um guia que descreve a soma de conhecimentos inerente referentes à Engenharia de Software, ou seja, abrange práticas já aprovadas e de uso comum.



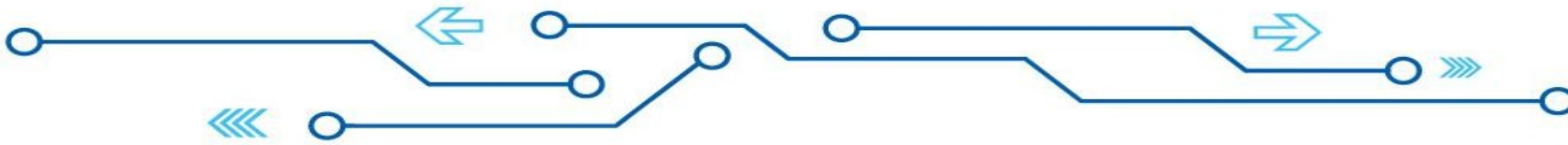
Objetivos do SWEBOK

- Promover uma visão consciente da engenharia do software no mundo inteiro;
- Esclarecer o lugar "e estabelecer o limite" da engenharia de software com respeito a outras disciplinas como ciência da computação, gerenciamento de projetos, engenharia da computação e matemática;
- Caracterizar os conteúdos da disciplina de engenharia de software;
- Fornecer um acesso ao Corpo de Conhecimento de Engenharia de Software;
- Fornecer uma base para desenvolvimento de currículo, certificação individual e licenciamento de material.



Áreas de Conhecimento

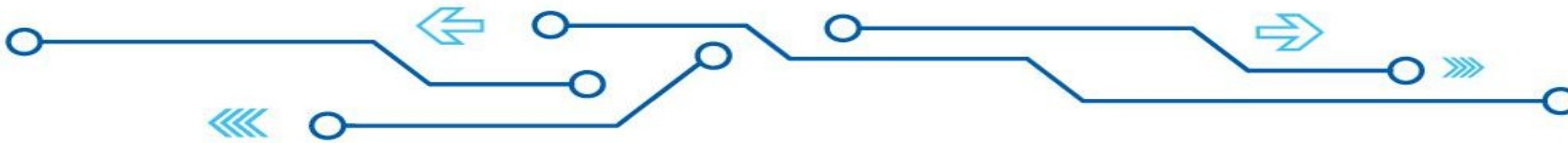
- Requisitos de Software;
- Design de Software;
- Construção de Software;
- Teste de Software;
- Manutenção de Software;
- Gerência de Configuração de Software;
- Gerência da Engenharia de Software;
- Processo de Engenharia de Software;
- Ferramentas e Métodos da Engenharia de Software;
- Qualidade de Software.



Requisitos de Software

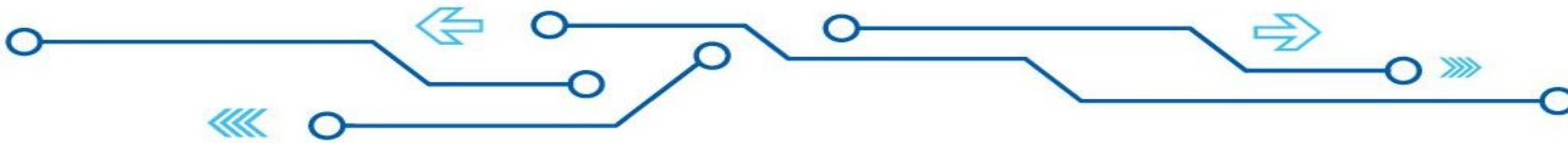
- Um requisito é definido como uma propriedade que deve ser exposta para resolver algum problema do mundo real;

- Essa área é subdividida em mais 7 subáreas:
 - Fundamentos de Requisitos de Software;
 - Processo de Requisitos;
 - Elicitação de Requisitos;
 - Análise de Requisitos;
 - Especificação de Requisitos;
 - Validação de Requisitos;
 - Considerações Práticas.



Requisitos de Software

- Fundamentos de Requisitos de Software:
 - Definição de requisito de software – propriedade que deve ser apresentada pelo software para resolver um problema do mundo real;
 - Requisitos de produtos e processo – o primeiro são as regras de negócio do software e segundo são as restrições de desenvolvimento impostas para a criação do software;
 - Requisitos funcionais e não funcionais – o primeiro descreve as funções que o sistema deve executar e o segundo são restrições ou requisitos de qualidade;
 - Requisitos quantificáveis – requisitos precisam ser definidos com clareza e sem ambiguidade e quando for o caso quantitativamente;
 - Requisitos de sistema e requisitos de software – o primeiro corresponde as necessidades para um sistema como um todo e o segundo é derivado do primeiro;



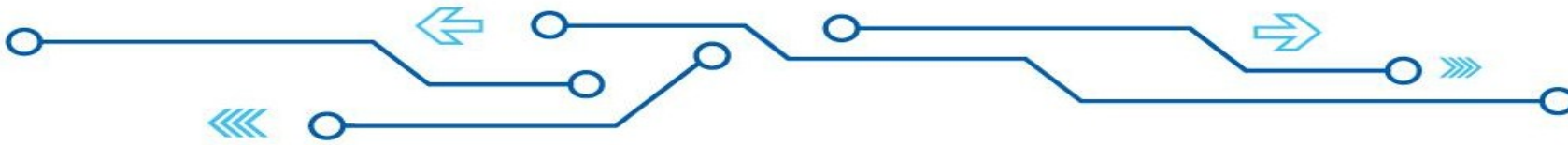
Requisitos de Software

- Processo de Requisitos:
 - Modelos de processo – identifica os requisitos de software como itens de configuração e os gerencia usando as práticas de GC;
 - Atores de processo – define os papéis dos participantes do processo;



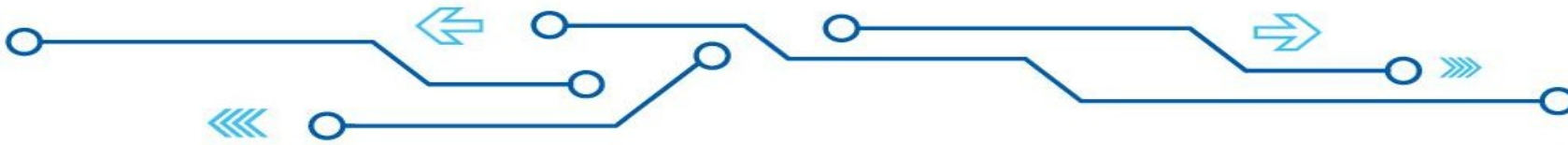
Requisitos de Software

- Elicitação de Requisitos:
 - Fontes de requisitos – objetivo do negócio, conhecimento do domínio da aplicação, pontos de vista dos diferente tipos de stakeholders;
 - Técnicas de elicitação – entrevistas, cenários, protótipos, reuniões com facilitadores, observação entre outros;



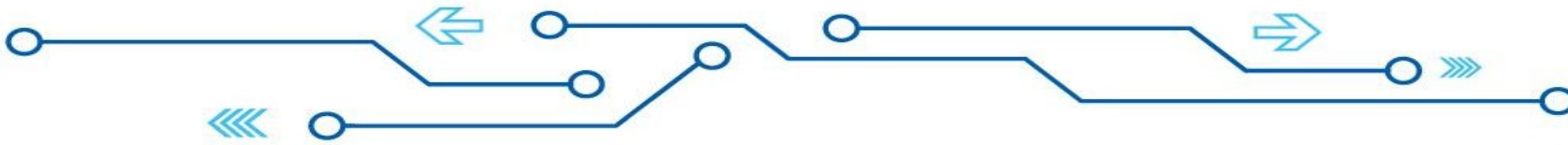
Requisitos de Software

- Análise de Requisitos:
 - Classificação de requisitos – por exemplo se o requisito é funcional ou não funcional ou a prioridade do requisito;
 - Modelagem conceitual – desenvolvimento de modelos de um problema do mundo real para auxiliar no entendimento deste;
 - Projeto de arquitetura e alocação de requisitos – é o ponto onde o processo de requisitos sobrepõe-se com o projeto de software ou sistema;
 - Negociação de requisitos – resolver conflitos entre requisitos.



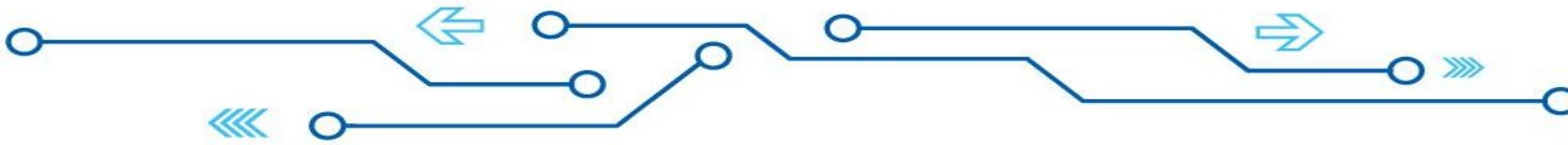
Requisitos de Software

- Especificação de Requisitos:
 - Documento de definição do sistema – registra em alto nível os requisitos do sistema;
 - Especificação de requisitos do sistema – aplica-se a sistemas que possuem número considerável de componentes;
 - Especificação de requisitos de software – define o que o produto deve ser e o que ele não deve ser, estabelecendo base para acordo entre os stakeholders;



Requisitos de Software

- Validação de Requisitos:
 - Revisões de requisitos – busca por erros, contradições, falta de clareza e desvios de práticas padrões;
 - Prototipagem – meio pelo qual é possível validar a interpretação que o engenheiro faz do requisito;
 - Validação do modelo – é necessário validar a qualidade dos modelos desenvolvidos durante a análise;
 - Testes de aceitação – o produto deve ser validado nos seus requisitos.



Requisitos de Software

- Considerações Práticas:
 - Natureza interativa do processo de requisito;
 - Gerência de mudança;
 - Atributos de requisitos;
 - Rastreamento de requisitos;
 - Medição de requisitos.



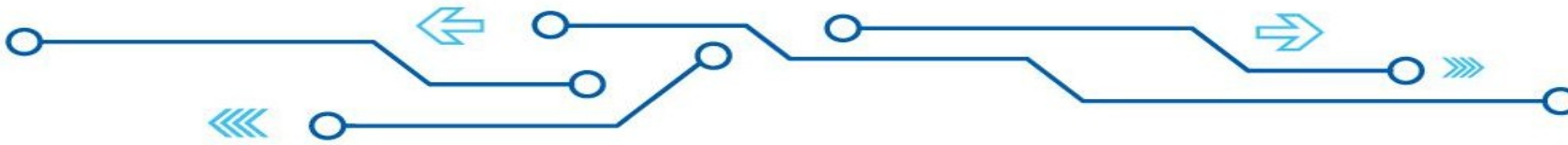
Design de Software

- Design é o processo de definir a arquitetura, componentes, interfaces, e outras características de um sistema ou componente;
- Essa área é subdividida em mais 6 subáreas:
 - Fundamentos de Design de Software;
 - Questões-Chaves em Design de Software;
 - Estrutura e Arquitetura de Software;
 - Análise de Qualidade do Design e Avaliação do Software;
 - Notações de Design de Software;
 - Estratégias de Design de Software e Métodos.



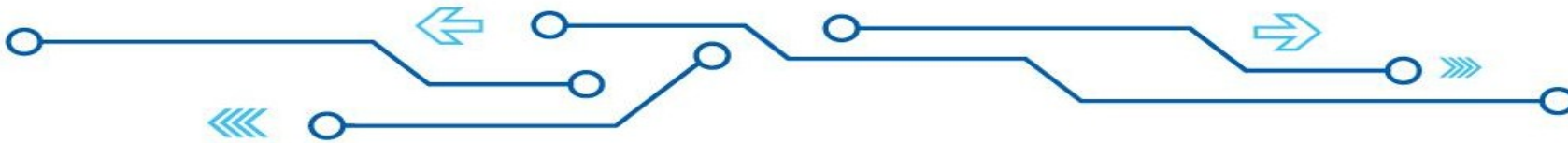
Design de Software

- Fundamentos de Design de Software:
 - Conceitos gerais de design – é uma forma de se resolver problemas, pois oferece objetivos, restrições, alternativas, representações e soluções;
 - Contexto do design de software – ligado fortemente ao contexto em que se enquadra o ciclo de vida de software
 - Processo de design de software – dividido em design arquitetural e design detalhado. O primeiro descreve como o software é composto e organizado dentro de componentes e o segundo o comportamento desses componentes;
 - Técnicas ativas – são os princípios do design de software, ou seja, noções chaves consideradas fundamentais no desenvolvimento do software.



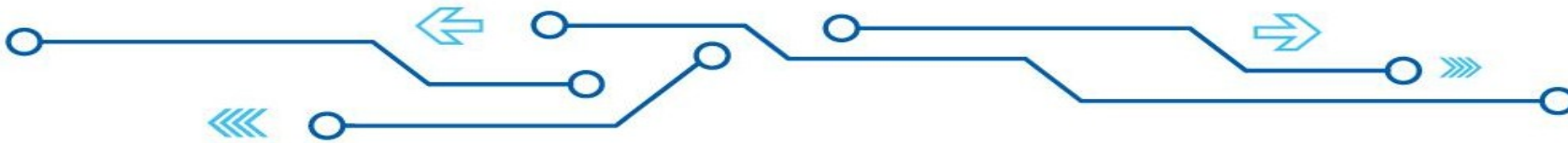
Design de Software

- Questões-Chaves em Design de Software:
 - Concorrência – como decompor o software em processos, tarefas e serviços e desejar eficiência, sincronização entre outros;
 - Controle e transporte de eventos – como organizar dados e controlar o fluxos de dados;
 - Distribuição de componentes – como distribuir o software baseado em hardware;
 - Transporte de exceção, erro e tolerância a falha – como prevenir e tolerar falhas;
 - Interação e apresentação – como estruturar e organizar interações com usuários e apresentação de informações;
 - Persistência dos dados – como dados de longa vida podem ser transportados.



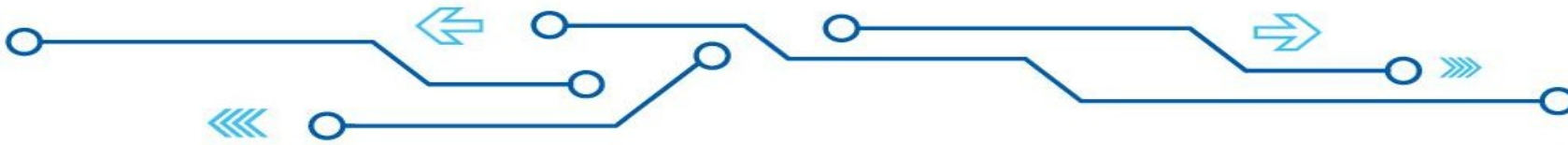
Design de Software

- Estrutura e Arquitetura de Software:
 - Estruturas arquitetônicas e viewpoints – diferentes facetas de alto nível de design de software podem e devem ser descritos e documentados;
 - Padrões de design – corresponde a uma solução comum para um problema comum em determinado contexto;
 - Famílias de programas e frameworks – permiti a reutilização de design de softwares e de componentes;



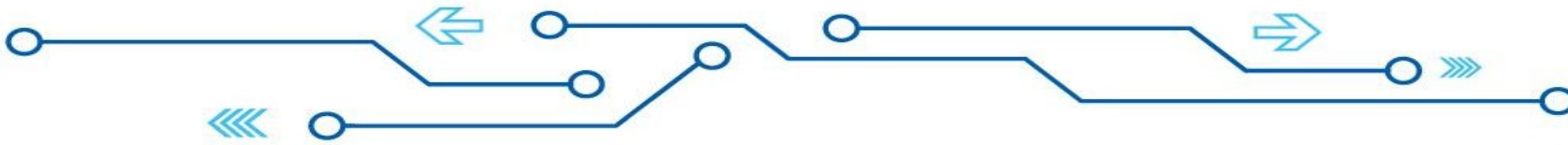
Design de Software

- Análise de Qualidade do Design e Avaliação do Software:
 - Atributos de qualidade – proporcionam a obtenção de manutenibilidade, portabilidade, testabilidade, rastreabilidade, correção entre outros;
 - Análise de qualidade e técnicas de avaliação – é a revisão do design de software de forma formal ou informal permitindo a avaliação do projeto;
 - Medidas – utilizadas para avaliar quantitativamente o tamanho do design de software, a estrutura ou a qualidade;



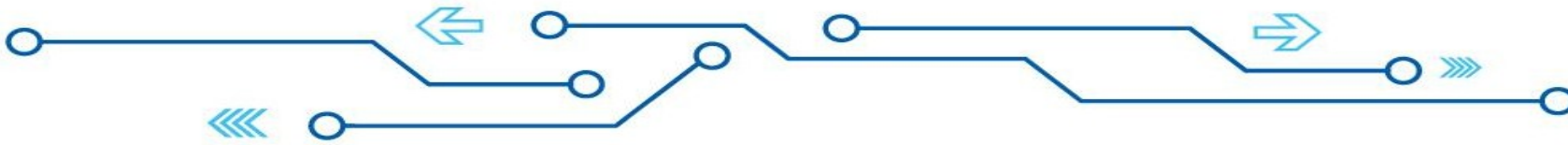
Design de Software

- Notações de Design de Software:
 - Descrições estruturais – fornece uma visão estática, isto é, descreve os componentes principais e como eles são interligados;
 - Descrições comportamentais – fornece uma visão dinâmica, isto é, descreve o comportamento dinâmico entre componentes de software;



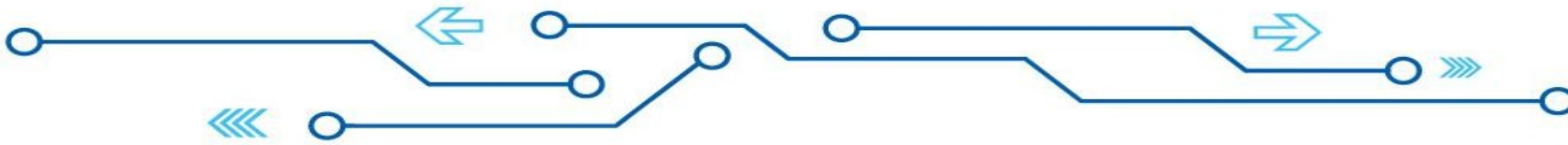
Design de Software

- Estratégias de Design de Software e Métodos:
 - Estratégias gerais;
 - Design orientado por função;
 - Design orientado a objeto;
 - Design centrado por estruturas de dados;
 - Design baseado em componente;



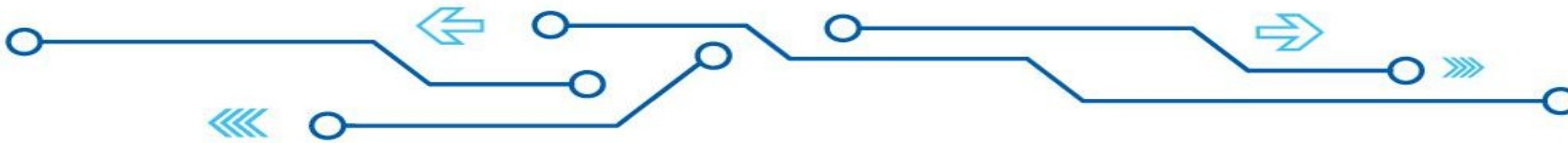
Construção de Software

- Se refere à criação detalhada de trabalho, ou seja, criação do software através da combinação de codificação, verificação, testes de unidades, testes de integração e depuração.
- Essa área é subdividida em mais 3 subáreas:
 - Fundamentos de Construção de Software;
 - Gerenciamento da Construção;
 - Considerações Práticas.



Construção de Software

- Fundamentos de Construção de Software:
 - Minimizar complexidade – permitir que as pessoas realizem tarefas complexas diminuindo suas limitações;
 - Antecipação de mudança – a maior parte dos softwares irá mudar com o tempo;
 - Construção para verificação – meio de construir software de forma que as falhas podem ser prontamente encontradas pelo engenheiro de software;
 - Normas de construção – envolvem o método de comunicação, linguagem de programação, plataforma, ferramentas, bem como normas internas da corporação;



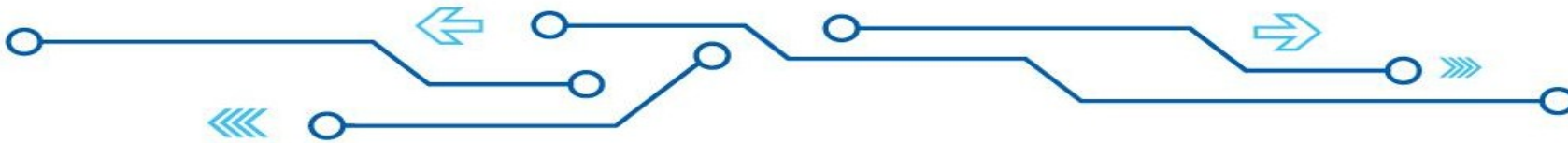
Construção de Software

- Gerenciamento da Construção:
 - Modelos de construção – escolha de qual modelo adotar;
 - Planejamento de construção – define a ordem em que os componentes são criados e entregues de acordo com o modelo escolhido;
 - Medição de construção;



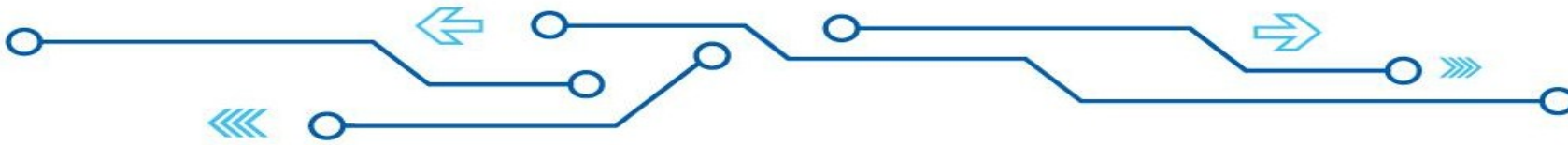
Construção de Software

- Considerações Práticas:
 - Design de construção – alguns detalhes do design ocorrerão em nível de construção;
 - Linguagem de construção – incluem todas as formas de comunicação em que um homem pode especificar uma solução de um problema executável para o computador;
 - Codificando – técnicas para criar o código fonte de forma compreensível, incluindo nomeações e layout;
 - Testando a construção – são divididos em teste de unidade e teste de integração;
 - Reutilização;
 - Qualidade da construção;
 - Integração.



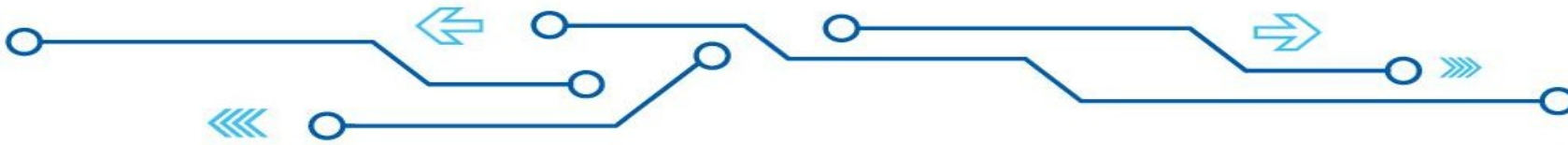
Teste de Software

- Compõem-se da verificação dinâmica de uma seleção de domínios de execuções normalmente infinito, contra o comportamento esperado.
- Essa área é subdividida em mais 5 subáreas:
 - Fundamentos de Teste de Software;
 - Níveis de Teste;
 - Técnicas de Teste;
 - Mensurações Relacionadas aos Testes;
 - Processo de Testes.



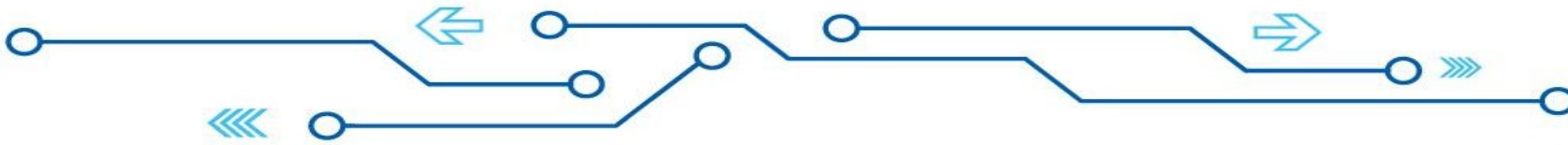
Teste de Software

- Fundamentos de Teste de Software:
 - Definir critérios de seleção de testes;
 - Objetivos do teste;
 - Teste para identificação de defeito;
 - Limitações teóricas e práticas de teste;
 - Testabilidade;
 - Relação do teste com outras atividades.



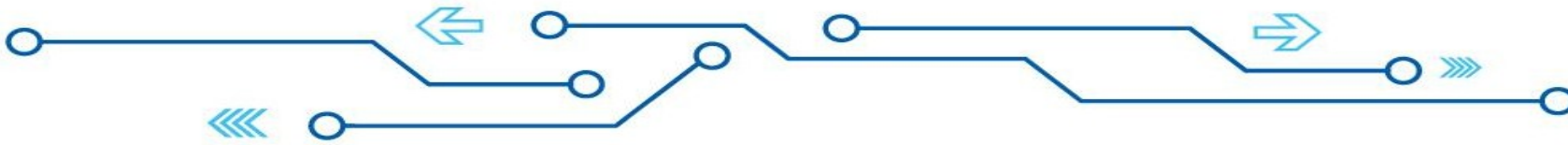
Teste de Software

- Níveis de Teste:
 - Alvo do teste (unidade, integração ou sistema);
 - Objetivos de teste (aceitação e qualificação, instalação, alfa e beta entre outros);



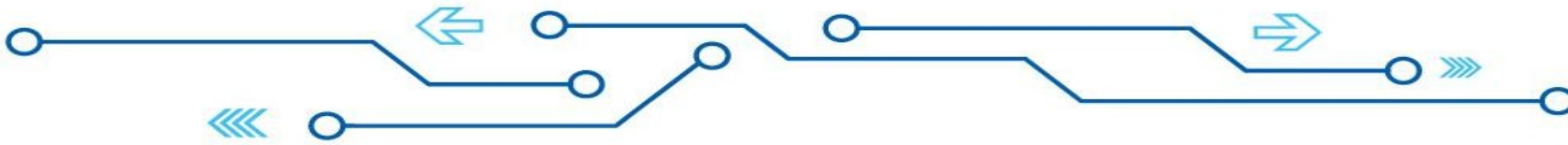
Teste de Software

- Técnicas de Teste:
 - Baseada na intuição e na experiência do engenheiro de software;
 - Baseadas em especificação;
 - Baseadas em código;
 - Baseadas em falha;
 - Baseada em uso;
 - Baseadas na natureza da aplicação;
 - Seleção e combinação de técnicas.



Teste de Software

- Mensurações Relacionadas aos Testes:
 - Avaliação do programa sob o teste;
 - Avaliação dos teste executados;
- Processo de Testes:
 - Considerações práticas;
 - Atividades de teste;



Manutenção de Software

- Uma vez em operação, as anomalias são descobertas, modificações no ambiente operacional e novos requisitos dos usuários são expostos.
- Essa área é subdividida em mais 4 subáreas:
 - Fundamentos de Manutenção de Software;
 - Questões-Chave em Manutenção de Software;
 - Processo de Manutenção;
 - Técnicas para Manutenção.



Manutenção de Software

- Fundamentos de Manutenção de Software:
 - Definição e terminologia;
 - Necessidade da manutenção;
 - Custos da manutenção;
 - Evolução do Software;
 - Categorias de Manutenção – Corretivas, Adaptativas, Perfectivas e Preventivas.



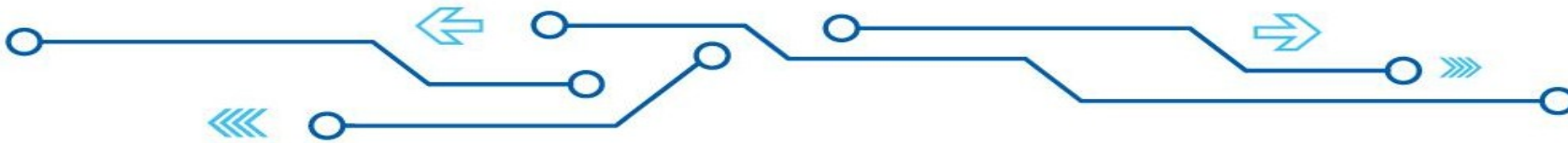
Manutenção de Software

- Questões-Chave em Manutenção de Software:
 - Problemas técnicos;
 - Gestão de problemas;
 - Estimativa de custo de manutenção;
 - Medição da manutenção do software;



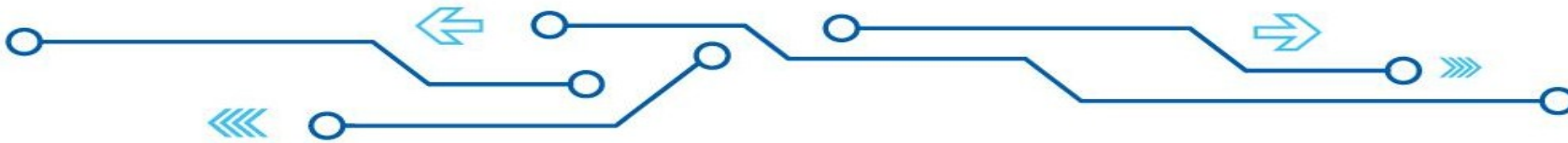
Manutenção de Software

- Processo de Manutenção:
 - Processo de manutenção;
 - Atividades de manutenção;
- Técnicas para Manutenção:
 - Compreensão;
 - Reengenharia;
 - Engenharia reversa;



Gerência de Configuração de Software

- Responsável por identificar a configuração do software em pontos distintos no tempo com o propósito de sistematicamente controlar modificações à configuração e de manter a integridade e a autoridade da configuração em todas as partes do ciclo de vida de sistema.
- Essa área é subdividida em mais 6 subáreas:
 - Gerência do Processo de GCS;
 - Identificação de Configuração de Software;
 - Controle de Configuração de Software;
 - Registros de Estado de Configuração de Software;
 - Auditoria de Configuração de Software;
 - Gerência de Liberação e Entrega de Software.



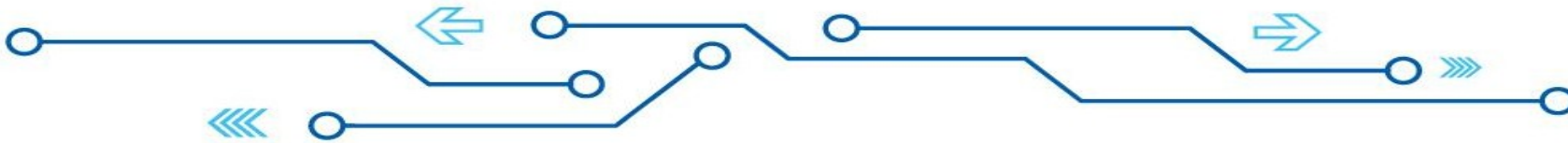
Gerência de Configuração de Software

- Gerência do Processo de GCS:
 - Contexto organizacional para o GCS;
 - Limitações e orientações para o processo de GCS;
 - Planejamento para GCS;
 - Plano GCS;
 - Monitoramento do GCS;



Gerência de Configuração de Software

- Identificação de Configuração de Software:
 - Identificação dos itens a serem controlados;
 - Biblioteca de software;
- Controle de Configuração de Software:
 - Requerente, avaliação e aprovação de alterações de software;
 - Implementação de mudanças de software;
 - Desvios e dispensas;



Gerência de Configuração de Software

- Registros de Estado de Configuração de Software:
 - Status de configuração de software;
 - Software para relatar o status de configuração;
- Gerência de Liberação e Entrega de Software:
 - Construindo software;
 - Software de gerenciamento de liberação;



Gerência de Configuração de Software

- Auditoria de Configuração de Software:
 - Software de auditoria de configuração funcional;
 - Software de auditoria de configuração física;
 - Auditoria de processo de uma baseline;



Gerência da Engenharia de Software

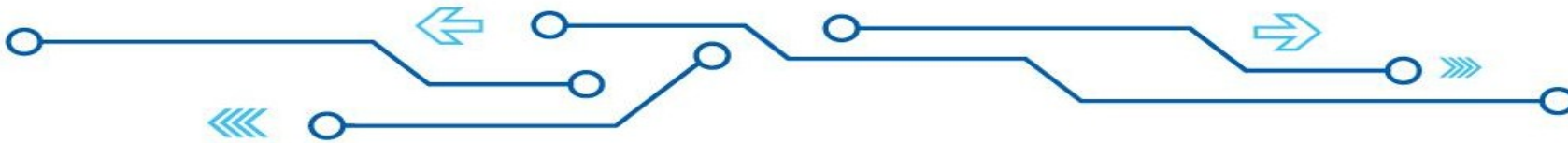
- Aponta o gerenciamento e mensuração da engenharia de software;
- Essa área é subdividida em mais 6 subáreas:
 - Iniciação e Definição de Escopo;
 - Planejamento de Projeto de Software;
 - Aprovação do Projeto de Software;
 - Revisão e Avaliação;
 - Fechamento;
 - Mensuração da Engenharia de Software.



Gerência da Engenharia de Software

Iniciação e Definição de Escopo

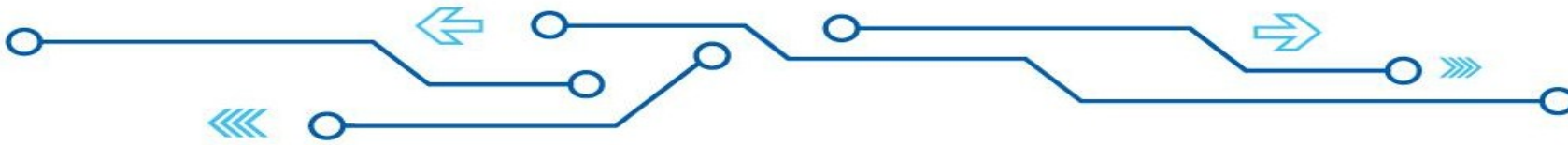
- Determinação e Negociação de Requisitos;
- Análise de Viabilidade;
- Processos para Análise e Revisão de Requisitos.



Gerência da Engenharia de Software

Planejamento de Projeto de Software

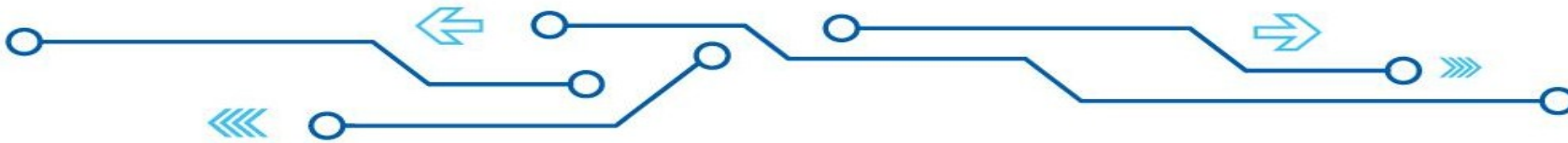
- Planejamento do Processo;
- Determinação de Entregas;
- Estimativa de Esforço, Cronograma e Custo;
- Alocação de Recursos;
- Gerenciamento de Riscos e Qualidade;
- Gerenciamento do Planejamento.



Gerência da Engenharia de Software

Aprovação do Projeto de Software

- Implementação dos Planos;
- Gerenciamento de Contratação de Fornecedores;
- Implementação do Processo de Medição;
- Processo de Monitoramento;
- Controle do Processo;
- Relatórios.



Gerência da Engenharia de Software

Revisão e Avaliação

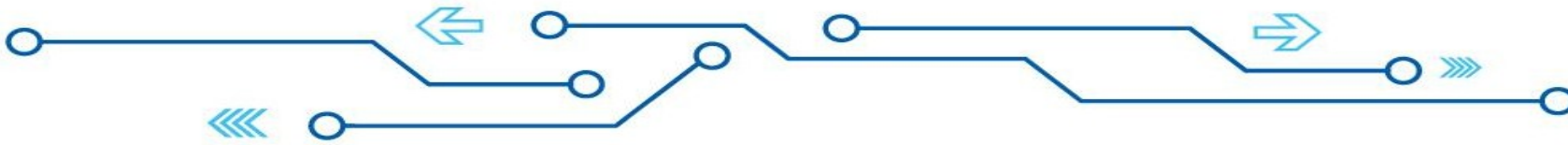
- Determinação da Satisfação dos Requisitos;
- Análise e Avaliação do Desempenho.



Gerência da Engenharia de Software

Fechamento

- Determinando o Término;
- Atividades de Fechamento.



Gerência da Engenharia de Software

Mensuração da Engenharia de Software

- Estabelecer as Medições;
- Planejamento do Processo de Medição;
- Execução do Processo de Medição;
- Avaliação das Medições.



Processo de Engenharia de Software

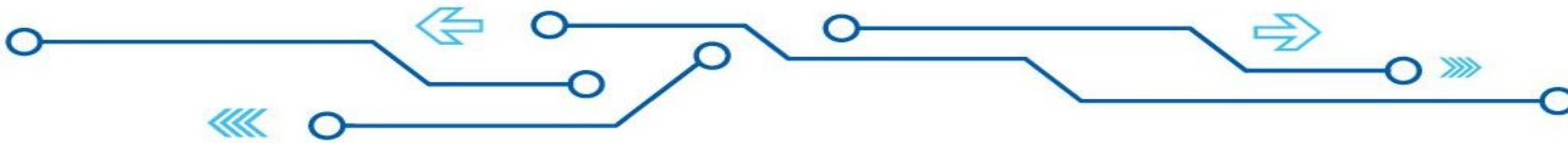
- Trata da definição, implementação, avaliação, mensuração, gerenciamento, alterações e melhoria do próprio processo de engenharia de software;
- Essa área é subdividida em mais 4 subáreas:
 - Processo de Implementação e Mudanças;
 - Definição do Processo;
 - Avaliação de Processo;
 - Mensuração de Produto e Processo.



Processo de Engenharia de Software

Processo de Implementação e Mudanças

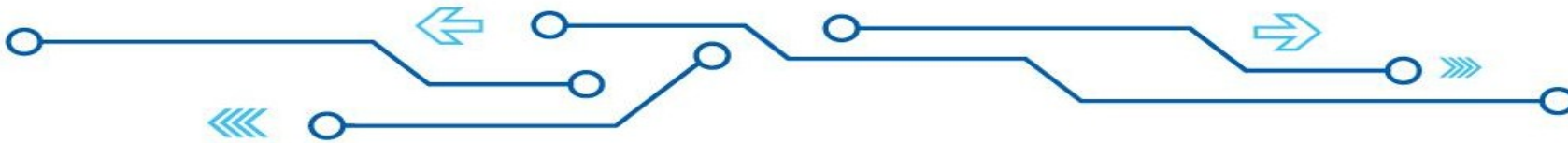
- Processo de Infra-Estrutura;
- Ciclo de Gerenciamento do Processo de Software;
- Modelos de Processo de Implementação e Mudanças;
- Considerações Práticas.



Processo de Engenharia de Software

Definição do Processo

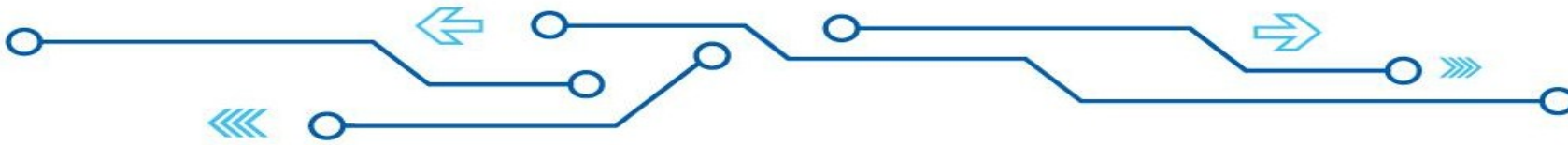
- Modelos de Ciclo de Vida de Software;
- Processos de Ciclo de Vida de Software;
- Notações para Definição de Processos;
- Adaptação de Processo;
- Automação;



Processo de Engenharia de Software

Avaliação de Processo

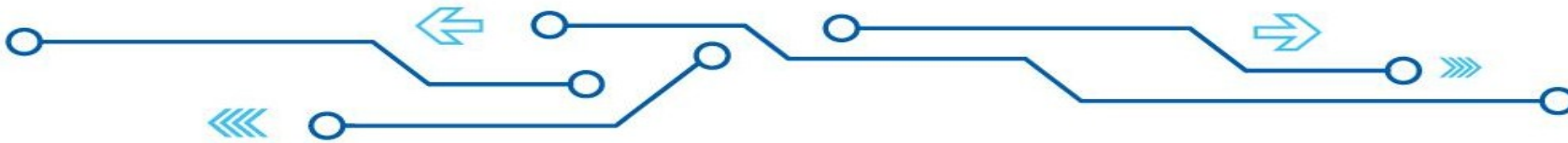
- Modelos de Avaliação de Processos;
- Métodos para Avaliação de Processos.



Processo de Engenharia de Software

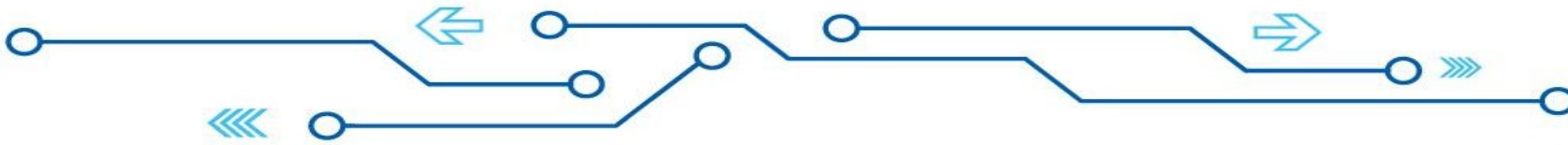
Mensuração de Produto e Processo

- Medição do Processo e do Produto de Software;
- Medição da Qualidade dos Resultados;
- Modelos de Informação do Software;
- Técnicas de Medição do Processo.



Ferramentas e Métodos da Engenharia de Software

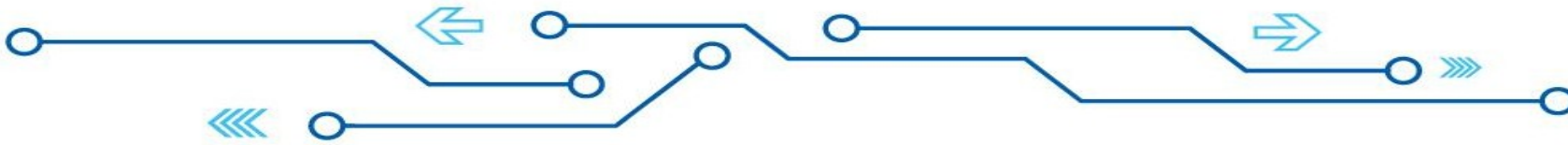
- Inclui ferramentas e métodos para serem aplicados na engenharia de software;
- Essa área é subdividida em mais 2 subáreas:
 - Ferramentas para Engenharia de Software;
 - Métodos para Engenharia de Software.



Ferramentas e Métodos da Engenharia de Software

Ferramentas para Engenharia de Software

- Ferramentas de Requisitos (modelagem e rastreabilidade);
- Ferramentas de Design;
- Ferramentas de Construção (Editores de Código, Compiladores, Interpretadores e Depuradores);
- Ferramentas de Teste (Geração, Execução, Avaliação, Gerenciamento, Análise e Desempenho);
- Ferramentas de Manutenção (Compreensão, Reengenharia e Engenharia Reversa);
- Ferramentas de GCS (Gerenciamento de Defeitos e Erros, Monitoramento, Versionamento e Correção);
- Ferramentas de GES (Planejamento e Acompanhamento, Riscos e Medição).



Ferramentas e Métodos da Engenharia de Software

Métodos para Engenharia de Software

- Métodos Heurísticos: Estruturado, Orientado a Dados, Orientado a Objetos e Domínio Específico;
- Métodos Formais: Especificação de linguagens e notações, Refinamento e Propriedades de Verificação/Confirmação;
- Métodos de Prototipagem: Estilo, Objetivo e Avaliação.



Qualidade de Software

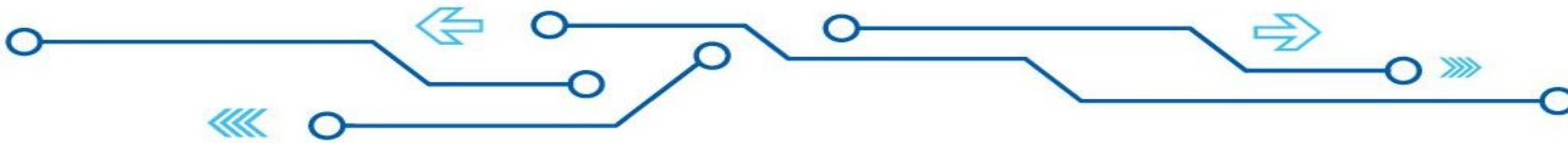
- Aborda considerações relativas à qualidade de software que vão além dos processos de ciclo de vida do software;
- Essa área é subdividida em mais 3 subáreas:
 - Fundamentos da Qualidade de Software;
 - Processos de Gerenciamento da Qualidade do Software;
 - Considerações Práticas.



Qualidade de Software

Fundamentos da Qualidade de Software

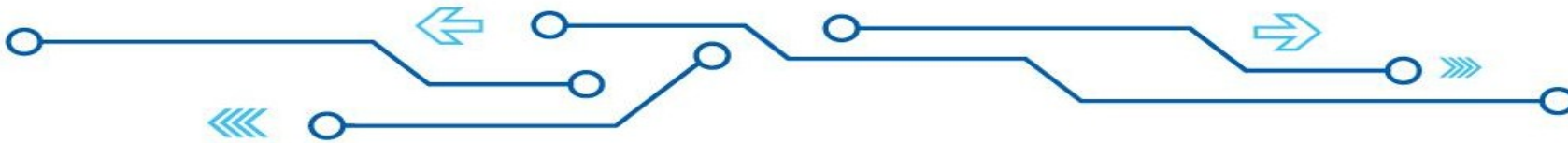
- Cultura e Ética da Engenharia de SW;
- Valor e Custo da Qualidade;
- Modelos e Características de Qualidade;
- Melhoria de Qualidade.



Qualidade de Software

Processos de Gerenciamento da Qualidade do Software

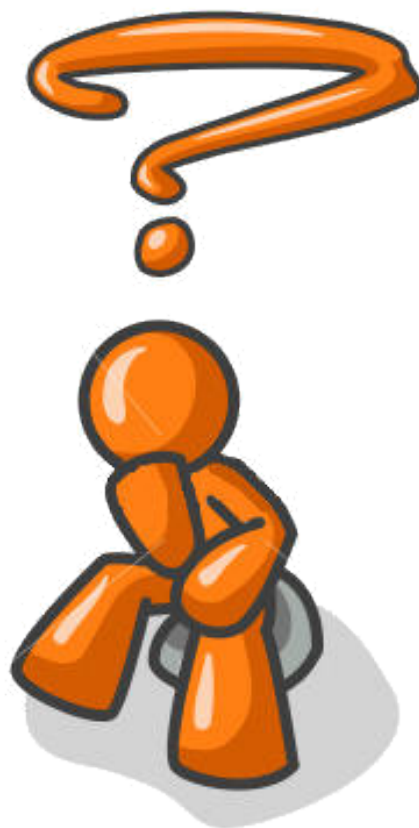
- Garantia da Qualidade;
- Verificação e Validação;
- Revisão e Auditoria.

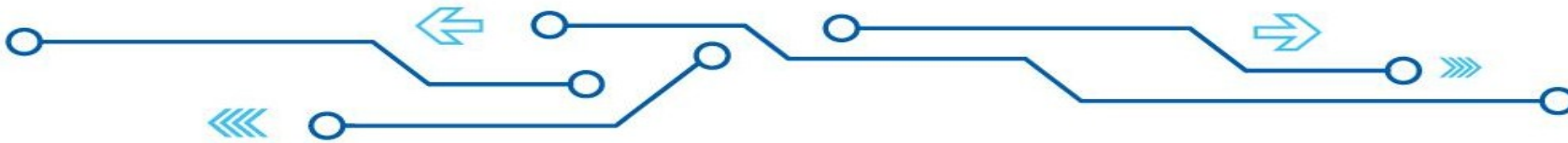


Qualidade de Software

Considerações Práticas

- Requisitos de Qualidade;
- Caracterização de Defeitos;
- Técnicas de Gerenciamento de Qualidade;
- Métrica de Qualidade.





Referência Bibliográfica

- **IEEE Computer Society.** Swebok - GUIDE to the software engineering body of knowledge. **Versão 2004.**