



Processo de Desenvolvimento

Processo de desenvolvimento de software

Vamos iniciar essa aula com a apresentação do conceito de processo de desenvolvimento de software, Sommerville define como: “Um processo de software é um conjunto de atividades relacionadas que levam a produção de um sistema de software” (SOMMERVILLE, 2019, pp 29)

Vamos apresentar alguns conceitos básicos sobre processo de desenvolvimento de software.

Portanto, o processo de desenvolvimento de software deve ter suas atividades bem definidas com a indicação precisa de quem irá realizar as atividades e com quais ferramentas essas atividades serão realizadas.

Usaremos a seguinte definição livre para o termo processo: é um conjunto de atividades realizado por pessoas podendo usar ferramentas.

Esta aula envolve as definições de ADIT, a engenharia de requisitos, e os ciclos de vida de desenvolvimento de software.

ADIT

O processo de desenvolvimento ADIT consiste em definir as fases, os papéis e as ferramentas, pois trata-se de um meta-modelo. Podemos destacar que é um processo trivial organizado em 4 fases organizadas da seguinte forma:

- fase 1 - Análise: indica as atividades para se analisar as necessidades e metas dos usuários e dessa forma realizar descoberta dos requisitos;

- fase 2 – Design: indica as atividades para organizamos a estruturas e arquiteturas de um sistema de software, aqui desc (🧑) nos as especificações;
- fase 3 - Implementação: indica as atividades realizadas para codificar o código que vai ser executado no ambiente de produção;
- fase 4 – Teste: indica as atividades para verificar as conformidades do aplicativo com as necessidades e metas dos usuários.

Requisito

Segundo Sommerville, os requisitos de um sistema são as descrições dos serviços que o sistema deve prestar e as restrições à sua operação. Esses requisitos refletem as necessidades dos clientes de um sistema que atende a um determinado propósito. (SOMMERVILLE,2019 pp 85)

Engenharia de Requisitos

A engenharia de requisitos define as atividades realizadas nos processos de descoberta, análise, documentação e conferência dos serviços e restrições de um sistema.

Na engenharia de requisitos são definidas as seguintes fases:

- 1) Elicitação;
- 2) Análise;
- 3) Especificação;
- 4) Validação.

Até esse ponto da aula vimos algumas definições triviais que são utilizadas em todas as metodologias de desenvolvimento de sistemas e softwares. Em continuidade a aula vamos utilizar uma divisão para organizar o entendimento sobre os ciclos de vidas de desenvolvimento de software. Sendo: Ciclo de vida tradicional para modelos antigos; E ciclo de vida ágil para os processos modernos utilizados n

Ciclo de vida Tradicional



Os ciclos de vidas tradicionais foram organizados para tentar resolver um problema que foi conhecido com uma crise do software na década de 70, na qual a gente teve um grande desenvolvimento do hardware da máquina do computador e o software ficou para trás. Dessa forma



Os processos mais conhecidos e definidos que nós podemos encontrar na literatura. São: cascata (waterfall); incremental; espiral; e o processo unificado da Rational (RUP).

Esses processos foram os percursos na organização do desenvolvimento dos projetos de sistemas.

Ciclo de Vida Ágil

Mesmo com a grande melhora na arte de desenvolver sistemas com os ciclos tradicionais, os clientes e sistemas ainda tinham muitos problemas. Então no final da década de 90 um grupo de engenheiros da indústria do software se reuniram para atacar o problema da necessidade de desenvolvimento rápido e lidar com a mutabilidade dos requisitos, surgiu o movimento ágil e divulgaram o manifesto ágil.

O manifesto ágil enumera diretivas e princípios para produzir software útil e de maneira rápida. Podemos destacar que as mudanças de requisitos são bem-vindas; a entrega de software funcionando como medida de progresso e o cliente envolvido conjuntamente no processo de desenvolvimento.


Vários modelos ágeis foram desenvolvidos e destacamos:

- 1) Scrum;
- 2) XP programação extrema
- 3) Lean startup

Atividade Extra

Leituras complementares



Artigo de Edsger Dijkstra sobre a crise do software (DIJKSTRA, 1972)
Disponível 
em <https://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF> ,
acessado em 10/07/2021

Leitura do Capítulo 2 – Processos de Software do livro de Engenharia de Software do autor Ian Sommerville (qualquer edição)

Referência Bibliográfica

Crise do software , 29 set. 2020. (Nota técnica).

DIJKSTRA, E. The humble programmer. ACM Turing Lecture 1972, n. ACM 15, p. 1–15, 1972.

SOMMERVILLE, I. Engenharia de Software. Tradução: Luiz Cláudio Queiroz. 10. ed. São Paulo: Pearson, 2019.

REINEHR, S. Engenharia de Requisitos. Editora Grupo A. 1ª Edição. 2020.

SOMMERVILLE, I. Engenharia de Software. Editora Pearson. 10a Edição.

Ir para questão

