

Lição 1



Introdução à Engenharia de Software – Parte 1

Objetivos

Ao final desta lição, o estudante será capaz de:

- Ter uma visão de camadas sobre a Engenharia de Software
- Obter qualidade dentro do esforço do desenvolvimento
- Conhecer as técnicas e garantias de Qualidade de Software
- Conhecer o processo de software
- Compreender o desenvolvimento de sistemas
- Definir as pessoas envolvidas no esforço do desenvolvimento
- Conhecer as diversas formas e benefícios de documentação

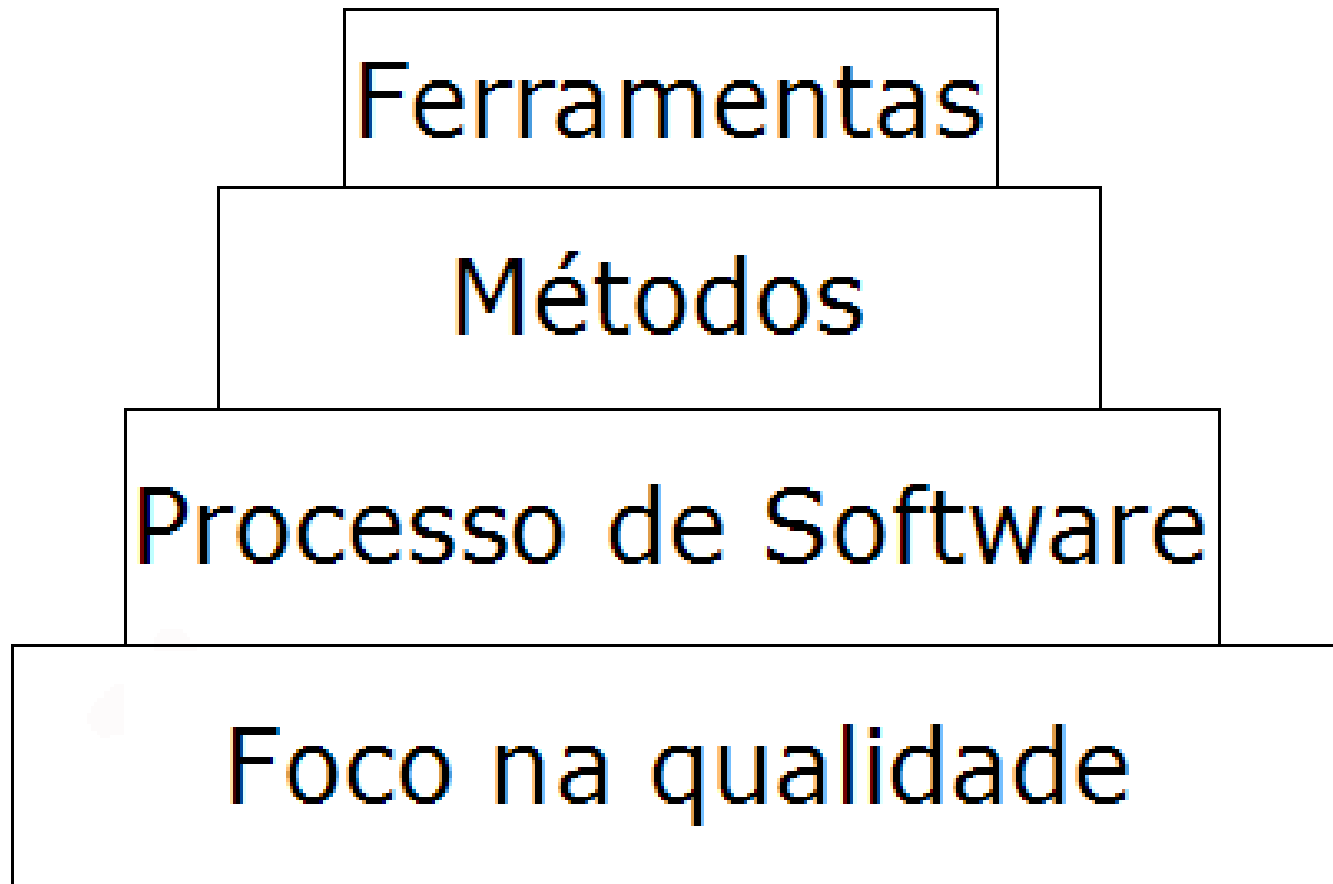


Engenharia de Software

- Fritz Bauer [NAU69]
 - Engenharia de Software é o estabelecimento e uso dos princípios da engenharia para obter economicamente software seguro e que trabalhe efetivamente em máquinas reais
- IEEE [IEEE93]
 - Engenharia de Software: (1) A aplicação sistemática, disciplinada, abordagem quantificável para o desenvolvimento, operação, e manutenção do software, que é, a aplicação da engenharia de Software. (2) O estudo de abordagens como em (1)



Camada de Visão



Foco na Qualidade

- Compromisso em melhoria continua no processo
- Permite o desenvolvimento de mais abordagens efetivas
- Define qualidade como fundamento do software

Processo de Software

- Define uma estrutura consistente da área chave do processo
- Integra outras camadas
- Estabelece:
 - Métodos técnicos aplicados
 - Ferramentas utilizadas
 - Produtos de trabalho produzidos
 - Marcos definidos
 - Técnicas asseguram qualidade



Métodos

- Proporciona a técnica *how-to* para construir software
- É uma sistemática, estabelecida, ou procedimento organizado ou caminho de fazer qualquer coisa
- Inclui uma gama ampla de tarefas, tais como, análise de requisitos, design, construção do programa, teste e manutenção

Exemplos de Metodologia

- Metodologias Estruturada:
Informações de Engenharia
Desenvolvimento do Ciclo de vida do Software/Projeto
Metodologia de Desenvolvimento de Aplicação Rapid
Metodologia de Desenvolvimento de Aplicação Joint
Método CASE*
- Metodologias Orientada a Objeto:
Método Booch
Método Coad e Yourdon
Método Jacobson
Método Rumbaugh
Método Wirfs-Brock



Ferramentas

- Fornece suporte automático ou semi-automático para processo e métodos
- Ferramentas de Engenharia de Software Computer-aided fornece um sistema de suporte a projeto de desenvolvimento de software, onde informações criadas por uma ferramenta pode ser usada por outra
- A maioria das ferramentas são usadas para desenvolver modelos

Modelo

- Tipos:
 - Merriam-Webster e Dicionário Garfield
 - Rational Rose
- O que modelamos
 - Modelo de Sistema
 - Modelo de Software



Por que modelamos?

- Focaliza-se nas características importantes do sistema
- Permite compreender melhor o sistema ou software

Exemplo de Ferramentas de Modelagem

- Abordagem Estruturada
 - Diagrama de Entidade-Relacionamento
 - Diagrama de Fluxo de Dados
 - Pseudocódigo
 - Fluxograma
- Abordagem Orientada a objeto
 - Linguagem de Modelagem Unificada (UML)



Qualidade

- Característica de uma entidade para satisfazer necessidades declaradas e implícitas
- Encontrar requisitos dos usuários, permitindo o que querem, quando querem, a todo o tempo
- Construir software que encontram as necessidades e requisitos do usuário que é entregar a tempo e dentro do orçamento



Três Perspectivas sobre Qualidade

- Qualidade do Produto
- Qualidade do Processo
- Qualidade no Contexto do Ambiente de Negócio

Endereçando Pontos Importantes sobre Qualidade

- Uso de padrões de Qualidade
- Compreender pessoas envolvidas no processo de desenvolvimento
- Compreender as tendências sistemáticas na natureza humana
- Requer compromisso da organização e todos os envolvidos
- Requisitos de usuários administradores que mudarão ao longo do tempo

Garantia de Qualidade de Software

- É um subconjunto da engenharia de software
- Assegurar que as entregas (software) estarão de concluídas e irão de acordo com os requisitos e padrões dos usuários
- Considerar uma atividade do primordial que seja aplicada durante todo o processo de desenvolvimento de software
- **Objetivo:** Detectar erros antes do software ser entregue como produto final para o usuário final, quem o usará diretamente



Garantia de Qualidade de Software

- Abrange:
 - maior qualidade na gerencia
 - tecnologia eficaz de engenharia de software (métodos e ferramentas)
 - técnicas formais de revisão
 - estratégia de teste dividida em multi-camadas
 - controle do software, documentação e alterações feitas
 - um procedimento para assegurar a conformidade com padrões de desenvolvimento de software
 - medida e mecanismo de relatório



Qualidade de Software

- Significa software ajustado para uso
- Encontrar os requisitos dos usuários
- Exigências explicitamente indicadas funcionais e de desempenho



Três Pontos Importantes

- 1. Requisitos de software são a base para se mensurar a qualidade
- 2. Padrões definem um conjunto dos critérios de desenvolvimento
- 3. Requisitos implícitos devem ser identificados e documentados

Características de Software Bem Projetado

- Usabilidade
- Portabilidade
- Reusabilidade
- Manutenibilidade
- Dependência
- Eficiência



Atividades SQA

- Preparar o plano de SQA para o projeto durante o planejamento
- Preparar o plano de SQL
- Participar no desenvolvimento da descrição do processo do projeto de software
- Rever as atividades da tecnologia de programação empregadas pela equipe do desenvolvimento para verificar se há a conformidade com o processo do desenvolvimento do software
- Revisões da equipe do SQA trabalham produtos para verificar se há a conformidade com os padrões definidos
- Assegurar de que os desvios nos produtos das atividades e do desenvolvimento de software estejam baseados em padrões definidos e operando-se em procedimentos
- Equipe do SQA relata desvios e não conformidades aos padrões à gerência sênior ou às partes interessadas.



Técnicas Formais de Revisão

- Atividade que garante a qualidade do software
- Objetivos são:
 - Descobrir erros
 - Verificar metas
 - Assegurar que o software obedece aos padrões
 - Garantir que o desenvolvimento será uniforme
 - Projetos de fácil manutenção

Método da Técnica Formal de Revisão

- Método de inspeção Fagan
- Revisão em pares

Método de Inspeção Fagan

- Procedimento de inspeção
 - As inspeções são realizadas em vários de pontos, no processo de planejamento do projeto e do desenvolvimento dos sistemas
 - Todas as classes com problemas na documentação deverão ser inspecionadas: não meramente logica, especificações e funções
 - A inspeção é realizada por colegas em todas as fases do desenvolvimento exceto pelo chefe
 - As inspeções são realizadas em uma lista prescrita das atividades
 - As reuniões da inspeção são limitadas a duas horas



Método de Inspeção Fagan

- Procedimentos de inspeção
 - As inspeções são conduzidas por um líder
 - São atribuídos papéis específicos a cada inspetor, de acordo com suas habilidades
 - Uma lista com questionário feito pelo inspetor serão usados para definir tarefas e aumentar a probabilidade de se achar defeitos
 - Os materiais serão inspecionados minuciosamente para se poder encontrar o máximo de erro possível
 - Estatísticas em tipos de erros são chaves, e usado para os relatórios que são analisados em uma maneira similar à análise financeira



Método de Inspeção Fagan

- Planejamento
 - O moderador deverá estar muito bem preparado para efetuar a inspeção. Decide:
 - Quem serão os inspetores e o que trabalho eles desempenharão
 - Quando e onde os inspeções ocorrerão
 - Quando distribuir os materiais para inspeção
- Visão Geral
 - 30 minutos de apresentação por inspetor
 - Pode ser ignorado se o assunto for muito familiar



Método de Inspeção Fagan

- Preparando
 - Os inspetores trabalham sozinho por aproximadamente 1-2 horas. Eles irão se concentrar nos papéis dados pelo moderador e descobrirão defeitos presumidos na documentação
- Reuniões
 - As reuniões deverão ser assistidas por todos os inspetores incluindo o moderador
 - O desenvolvedor do produto em questão estará normalmente presente para explicar e defende-lo
 - Nenhuma discussão é permitida, nem mesmo defeito
 - Uma lista de defeito será produzida



Método de Inspeção Fagan

- Ajuste
 - A lista de problemas deverá ser entregue a uma pessoa que irá solucioná-los
- Continuação
 - O moderador é a pessoa responsável por assegurar-se que os problemas encontrados serão bem solucionados
 - A exatidão dos ajustes serão avaliados e testados pelos outros inspetores
- Reunião casual da análise
 - Estas podem ser convocadas por qualquer inspetor para apresentar sua visão, mostrar erros ou apresentar melhorias



Walkthrough - Revisão em pares

- Esta é uma técnica de revisão menos formal que a inspeção
- O produto deverá ser revisado por uma equipe de 3 a 5 pessoas
- Um escrevente deve anotar a lista de ações

Manual de *Walkthrough*

- Nenhum gerente deve estar presente
- Enfatize que o *walkthrough* é para a detecção de erro e não correção deste
- Manter um constante interesse do grupo
- Nenhuma contagem ou atribuição de nota
- Criticar o produto não o desenvolvedor do projeto
- Sempre documentar a lista de ações

Atividades antes do *Walkthrough*

- O calendário do *walkthrough* deve ser um dia ou dois adiantados
- Distribuir apropriadamente os materiais para revisão
- Pedir para que os revisores façam dois comentários positivos e um negativo sobre o *walkthrough*

Atividades do *Walkthrough*

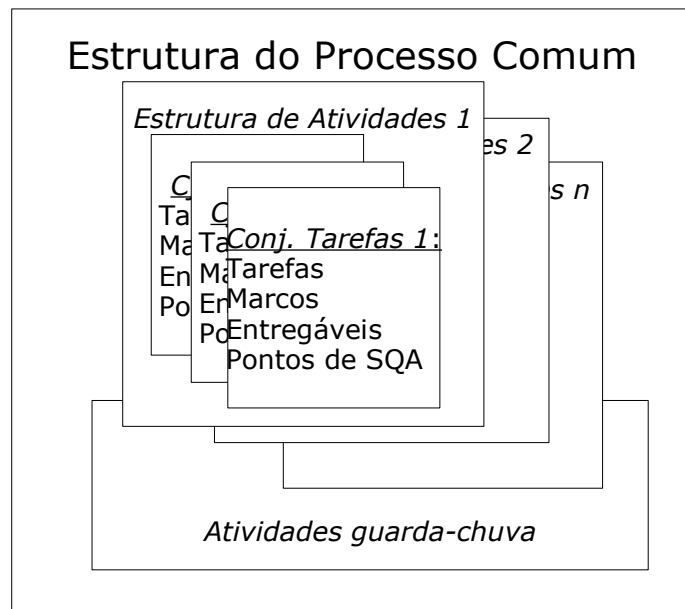
- O apresentador faz uma introdução breve do produto
- Solicite comentários dos revisores
- Edições são apresentadas e não resolvidas
- Não durar por mais de uma hora
- Uma lista de ações é o produto final



Atividades após o *Walkthrough*

- O gerente dá a lista da ação ao produtor do trabalho
- O produtor é questionado e deve submeter o relatório de status de ação para a resolução de erros ou discrepâncias apresentadas na lista de ação
- O produtor é responsável por enviar um relatório com as ações tomadas para resolver erros
- Um outro *walkthrough* deverá ser marcado

Processo de Software



- fornecer a estrutura a partir do qual um plano compreensivo de desenvolvimento de software que pode se estabelecer

Processo de Software

- Consiste de:
 - Uma estrutura de atividades
 - Um conjunto de tarefas
 - Atividades guarda-chuva

Final da Parte 1



- Continua...

Parceiros

- Os seguintes parceiros tornaram JEDITM possível em Língua Portuguesa:

