

Residência em TI Gerência de Configuração e Teste de Software

Prof. Eiji Adachi

Visão Geral

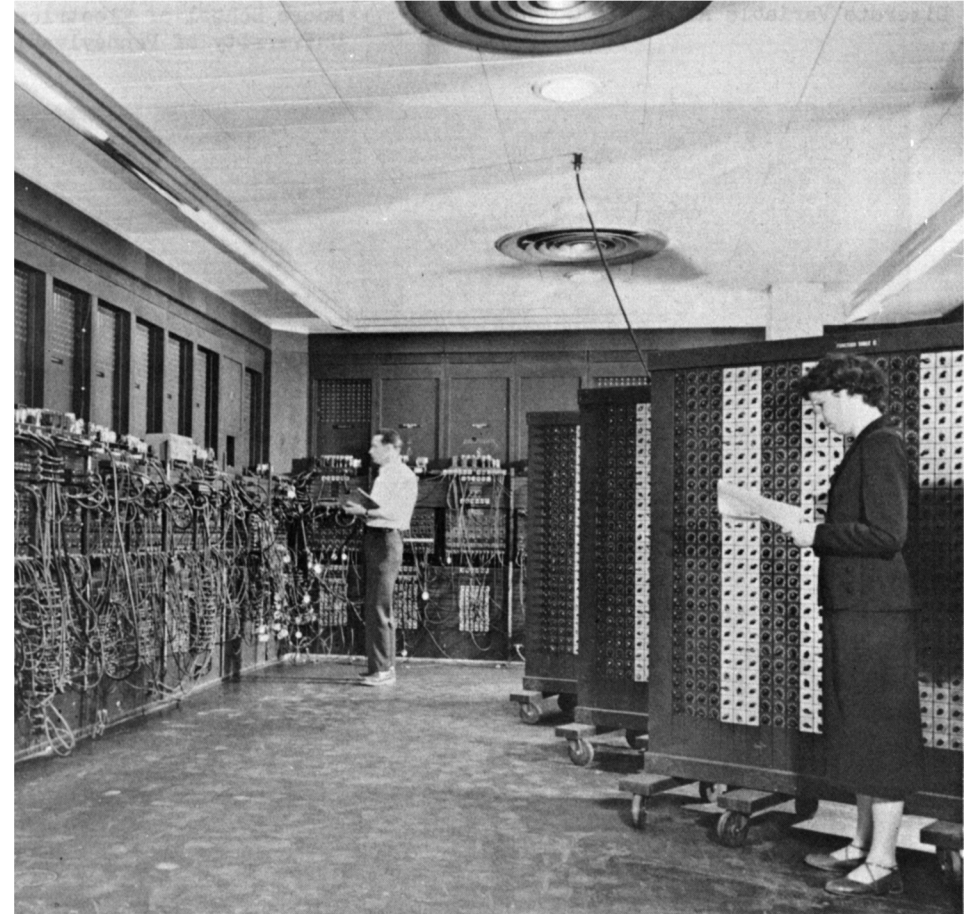
Visão Geral

- Antes de adentrarmos nos detalhes de Gerência de Configuração e de Teste de Software, vamos dar um passo atrás e ver um pouco de história e do contexto geral da Engenharia de Software para que possamos entender como essas duas disciplinas se encaixam num contexto maior de desenvolvimento de software

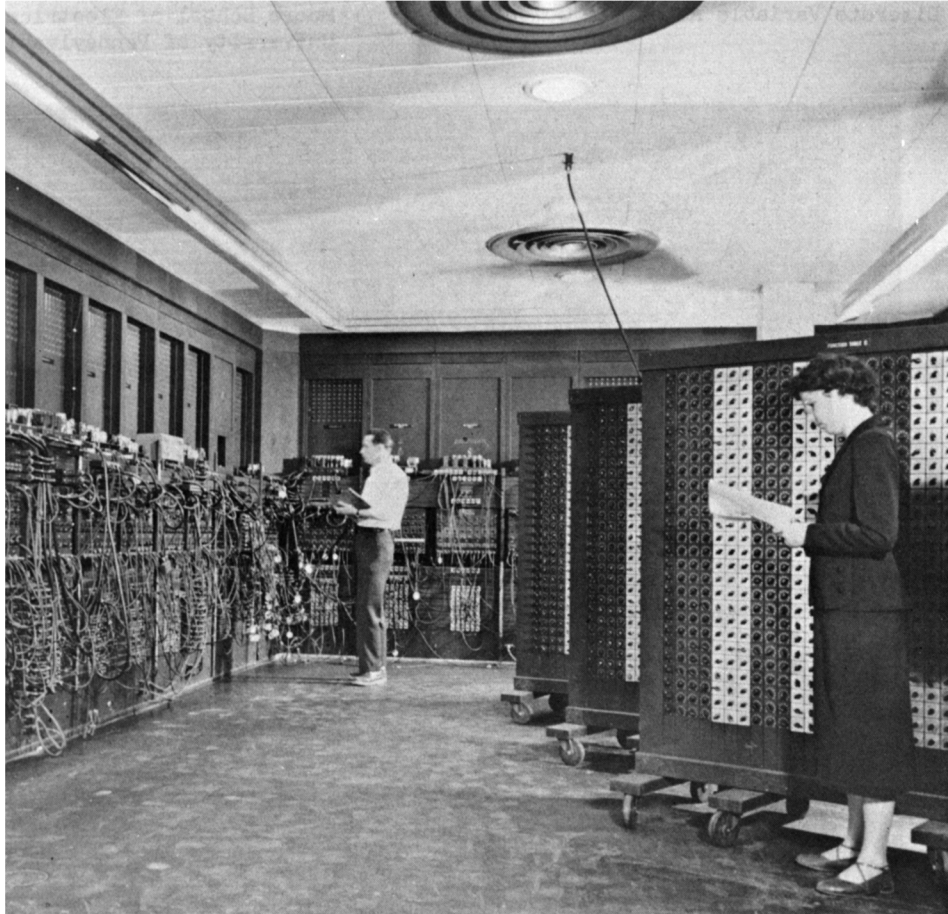
Breve História da Engenharia de Software

Anos 1950 e os Primeiros Computadores

- Computadores disponíveis em centros de pesquisa e universidades
 - 1957 – FORTRAN (Indústria - IBM)
 - 1958 – ALGOL (Academia)
- Computadores disponíveis para aplicações de negócios
 - 1959 - COBOL



Anos 1950 e os Primeiros Computadores



- Programas não eram complexos
 - Programas eram transcrições de soluções existentes para uma notação reconhecida por computadores
 - Computadores = “Calculadoras grandes e rápidas”

Anos 60

- Computadores (mainframes) popularizam-se e programas passam a (tentar) resolver problemas que ainda não tinham soluções
 - Problemas mais complexos



No Brasil

- Em 1957, o governo do Estado de São Paulo adquiriu um modelo Univac-120, mas apenas para ser usado no cálculo de consumo de água da capital.



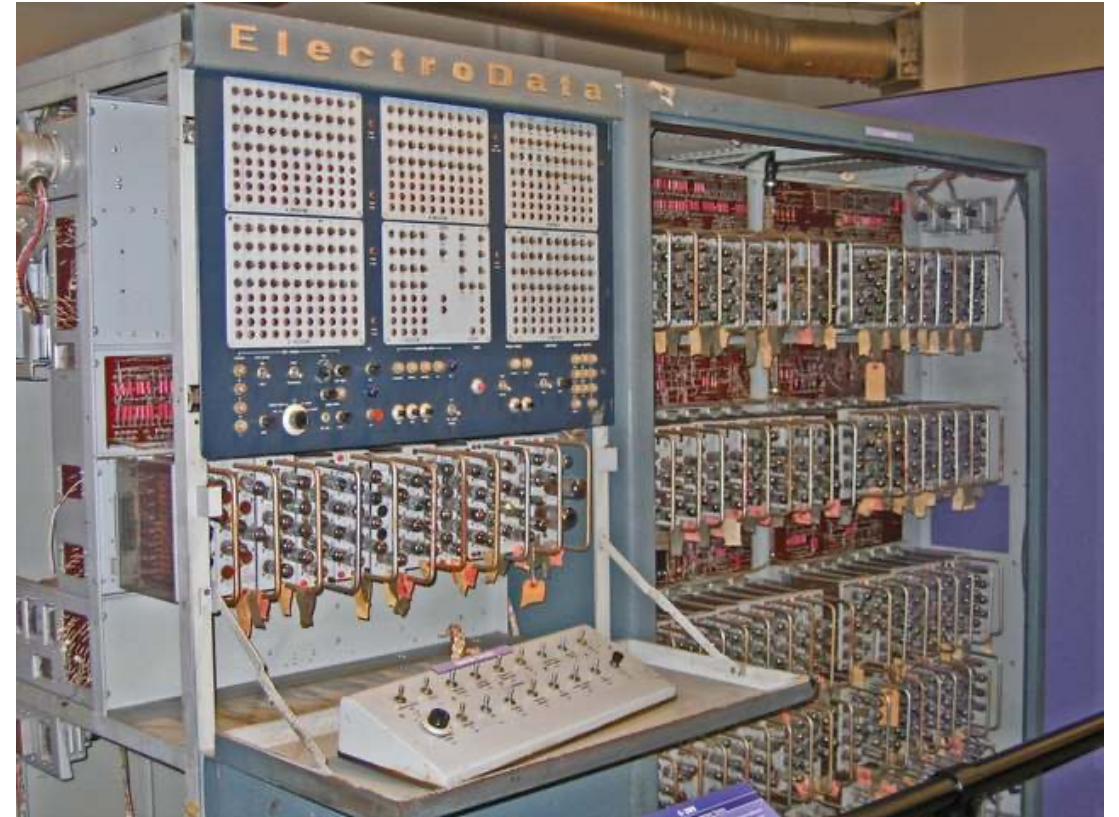
No Brasil

- Em 1959, no setor privado, a empresa Anderson Clayton (fábrica de óleo) instalou um Ramac 305 da IBM



No Brasil

- Junho de 1960, na PUC-Rio – o Burroughs Datatron B-205 chegou ao Brasil por um consórcio formado pelo CNPq, IPqM (Marinha) e Itamaraty
 - Memória de 4 mil palavras de 10 dígitos decimais, cerca de 16k.
 - Operações de adição, por exemplo, chegavam a consumir 0,1 milissegundos.



Anos 60 e a Crise do Software

- Projetos:
 - com orçamento estourado
 - com cronograma atrasado
 - difíceis de gerenciar
- Software:
 - ineficiente
 - de baixa qualidade
 - não atendia aos requisitos
 - difícil de manter

A principal causa da crise de software é que as máquinas se tornaram várias ordens de magnitude mais poderosas! (...) agora que temos computadores gigantescos, a programação se tornou um problema igualmente gigantesco.

Edsger Dijkstra



NATO Software Engineering Conference 1968



1968 – 1ª Conferência

- Working Conference on Software Engineering, sponsored by the NATO Science Committee
 - OTAN (NATO, no inglês) patrocinou conferência para se discutir os problemas no desenvolvimento de software

1968 – 1ª Conferência

- “A conferência foi assistida por mais de 50 pessoas, de 11 países diferentes, todas preocupadas profissionalmente com software, como usuários, fabricantes, ou professores universitários” [1]

[1] <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>

1968 – 1ª Conferência

- As discussões cobriram todos aspectos do software, incluindo:
 - Relação do software com o hardware
 - Design de software
 - Produção, ou implementação do software
 - Distribuição do software
 - Serviços no software

1968 – 1ª Conferência

- Também foram discutidos:
 - Como conseguir confiabilidade nos sistemas de dados que estão se tornando cada vez mais integrados às atividades centrais da sociedade moderna
 - Dificuldades de cumprir cronogramas e especificações em grandes projetos de software
 - Formação de engenheiros de software
 - Se o software deve ser precificado separadamente do hardware

1968 – 1ª Conferência

The basic problem is that certain classes of systems are placing demands on us which are beyond our capabilities and our theories and methods of design and production at this time.

1968 – 1ª Conferência

There are many areas where there is no such thing as a crisis — sort routines, payroll applications, for example. It is large systems that are encountering great difficulties. We should not expect the production of such systems to be easy.

1968 – 1ª Conferência

- Termos “Engenharia de Software” e “Crise do Software” se popularizaram
- Reconhecimento de que as práticas de desenvolvimento eram inadequadas

Engenharia de Software



Margaret Hamilton

“Eu lutei para trazer a legitimidade do software para que ele - e aqueles que o constroem - recebessem o devido respeito e assim comecei a usar o termo 'engenharia de software' para distingui-lo de hardware e outros tipos de engenharia, mas tratar cada tipo de engenharia como parte do processo geral de engenharia de sistemas.”

Engenharia de Software



Margaret Hamilton ao lado do software que ela e sua equipe produziram para o projeto Apollo

Margaret Hamilton foi diretora da Divisão de Software no Laboratório de Instrumentação do MIT, que desenvolveu o programa de voo usado no projeto Apollo 11, a primeira missão tripulada à Lua.

“Salvação” foi buscada em melhores metodologias e ferramentas...

...que levou ao surgimento da
Engenharia de Software como disciplina.

Engenharia de Software

Disciplina que objetiva aplicar
**Princípios de Engenharia para o
Desenvolvimento de Software.**

Princípios de Engenharia

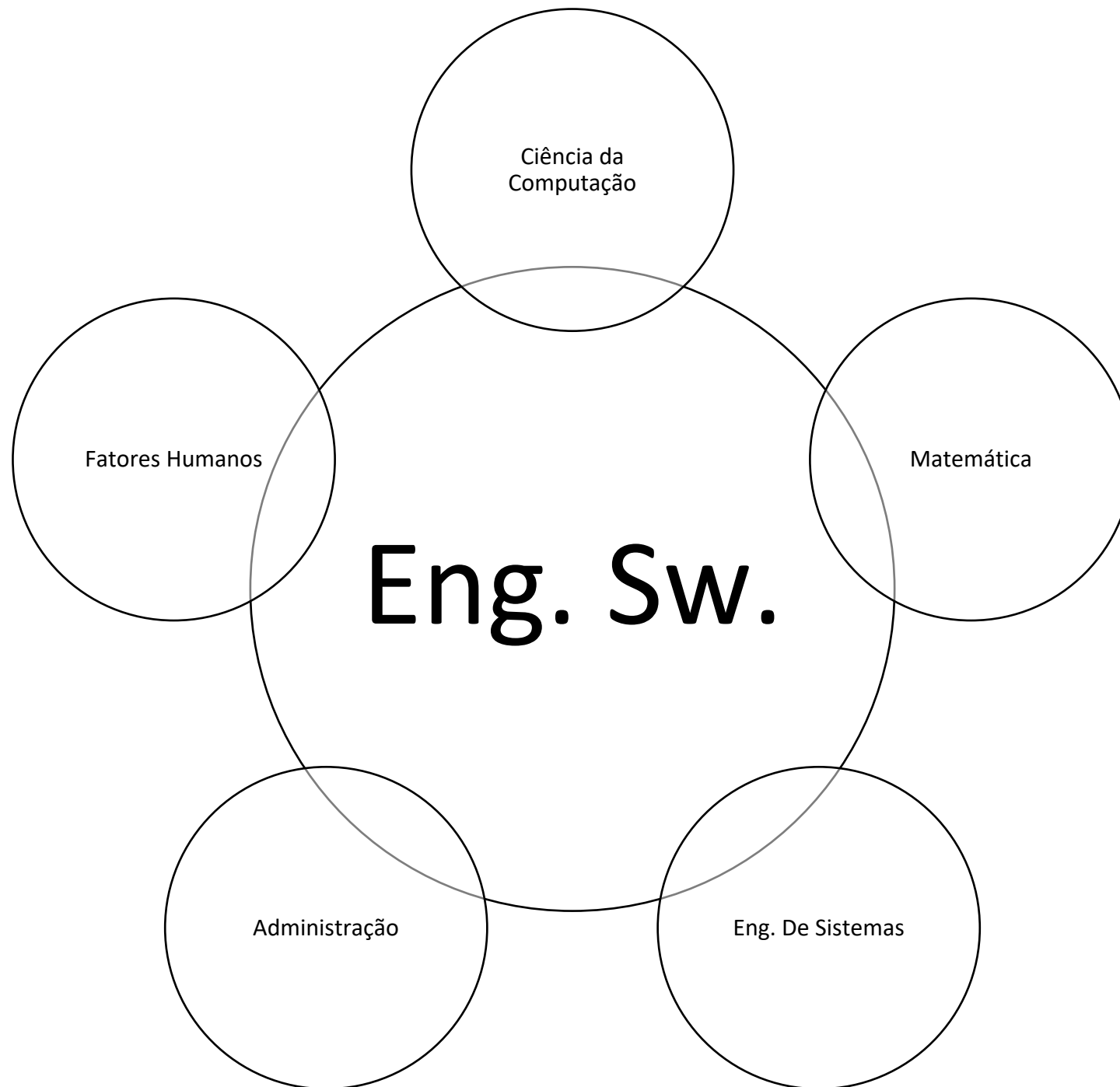
Aplicar conhecimentos científicos para resolver problemas do mundo real.

Engenharia de Software

Disciplina que objetiva aplicar **Métodos Sistemáticos e Disciplinados** para o **Desenvolvimento de Software.**

Engenharia de Software

Preocupa-se com todo o processo de
Desenvolvimento e Manutenção de
sistemas de software.



O que estuda a
engenharia de
software?

Grandes Áreas (lista não exaustiva)

- Requisitos
- Projeto e Arquitetura
- Construção
- **Teste**
- Manutenção
- **Gestão de Configuração**
- Gestão de Projeto
- Processos
- Modelos e Métodos
- Qualidade

Uma visão geral do desenvolvimento de software baseada em processos

- Processos Essenciais:
 - Requisitos, Arquitetura, Projeto Detalhado, Implementação, **Teste**, Manutenção, Operação, etc
- Processos de Apoio:
 - Gestão de Projeto, **Gerência de Configuração**, Garantia da Qualidade, Gestão de Riscos, Medição, etc

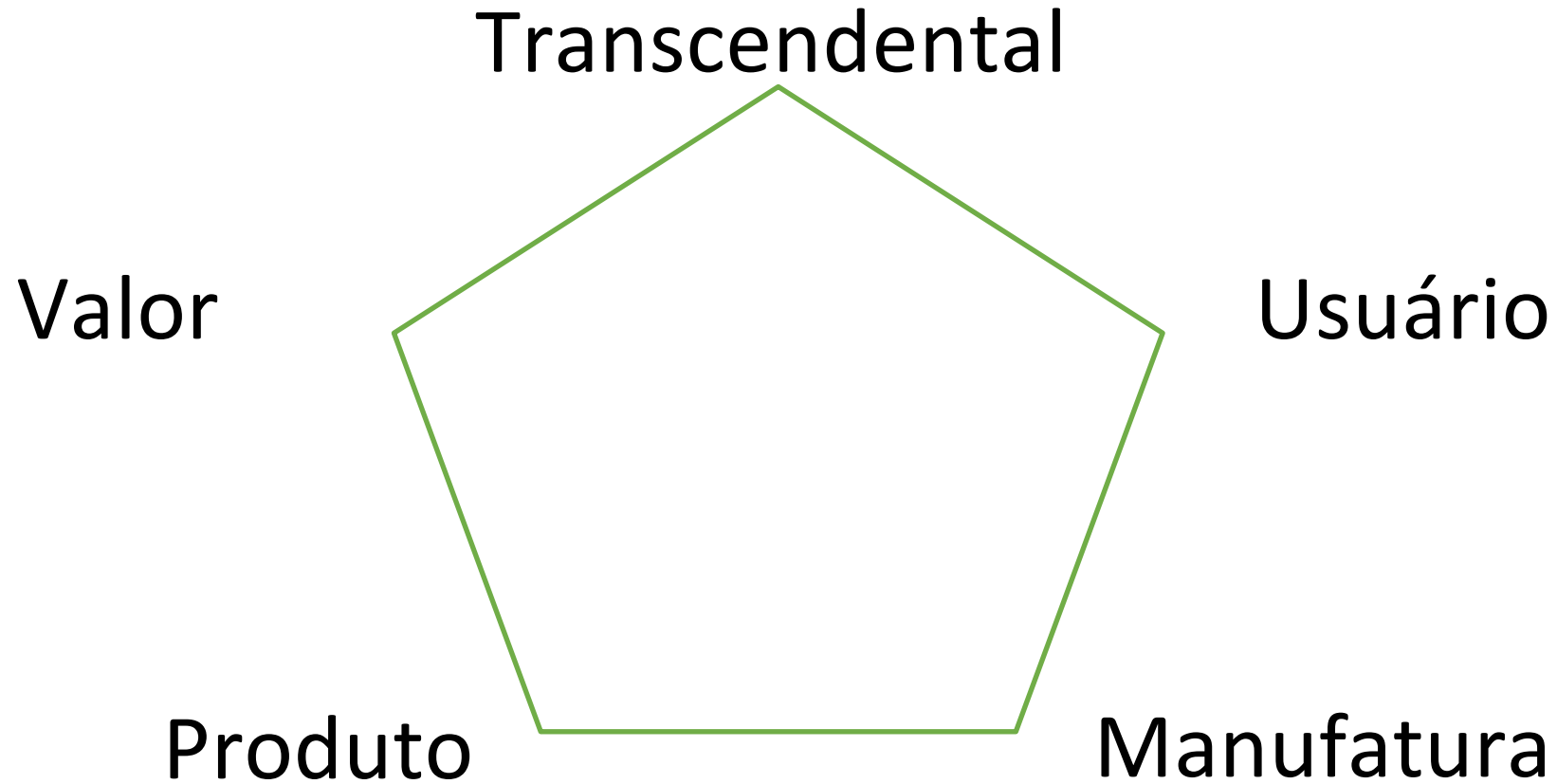
Visão Geral de Teste de Software

Engenharia e Qualidade de Software

- Engenharia de Software surge da dificuldade em se produzir software com qualidade

**Mas o que é
qualidade?**

Visões sobre Qualidade^[1]



[1] Kitchenham, Barbara, and Shari Lawrence Pfleeger. "Software quality: the elusive target [special issues section]." IEEE software 13.1 (1996): 12-21.

Visão Transcendental

*“Algo que se reconhece,
mas não se define”.*

Visão do Usuário

*“Aptidão (de um produto)
a um determinado fim
(necessidade do usuário)”.*

Visão da Manufatura

“Conformidade com a especificação”.

Visão do Produto

*“Características intrínsecas
do produto”.*

Visão do Valor

“O quanto um usuário está disposto a pagar por um produto”.

O que é
**qualidade de
software?**

Conjunto de características a serem satisfeitas num determinado grau de modo que o software satisfaça as necessidades de seus usuários.

Conjunto de características a serem satisfeitas num grau de modo que o software satisfaça as necessidades de seus usuários.

Visão do Produto

Conjunto de características a serem satisfeitas num determinado grau de modo que o software satisfaça as necessidades de seus usuários.

Visão da Manufatura

*Conjunto de características a serem satisfeitas num determinado modo que o software **satisfaça as necessidades de seus usuários.***

Visão do Usuário

*É como podemos
assegurar a
qualidade do
software?*

Gestão da Qualidade de Software

- Coleção de todos processos realizados para assegurar que os produtos de software atendem suas especificações de qualidade e satisfaçam seus usuários

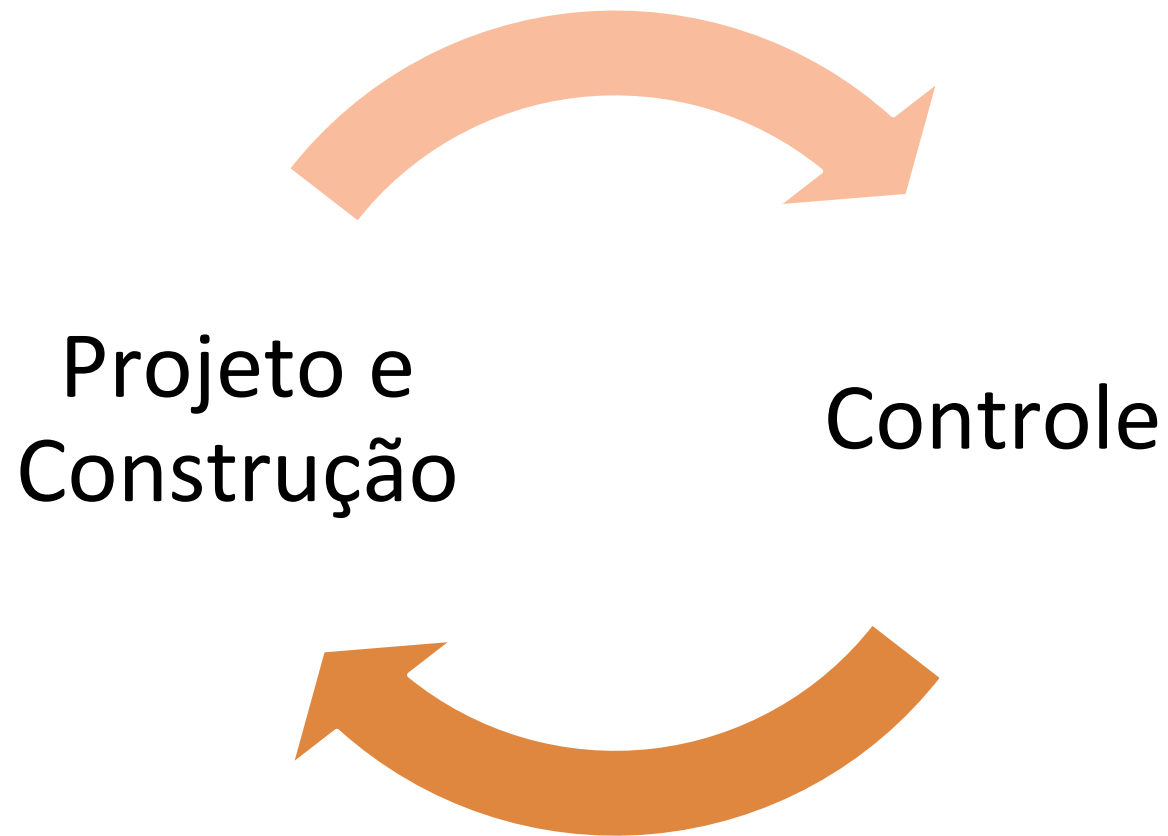
Gestão da Qualidade de Software

- Abrange 4 grandes classes de processos:
 - Planejamento
 - Garantia da qualidade
 - Controle de qualidade
 - Melhoria

Gestão da Qualidade de Software

- Abrange 4 grandes classes de processos:
 - Planejamento
 - Garantia da qualidade
 - **Controle de qualidade**
 - Testes de Software enquadram-se aqui!
 - Melhoria

Controle da Qualidade de Software



Controle da Qualidade de Software

- Disciplinas de Engenharia costumam parear atividades de Projeto e Construção com atividades de controle para checar produtos intermediários e finais para identificar e remover defeitos
 - Engenharia de Software não é diferente

Controle da Qualidade de Software

- Def.: Conjunto de procedimentos realizados para coletar evidências de que um produto de software atende seus requisitos de qualidade

Controle da Qualidade de Software

- Atividades de **Verificação** e **Validação (V&V)** são atividades de controle que objetivam prover evidências da existência de falhas nos produtos de software

Controle da Qualidade de Software

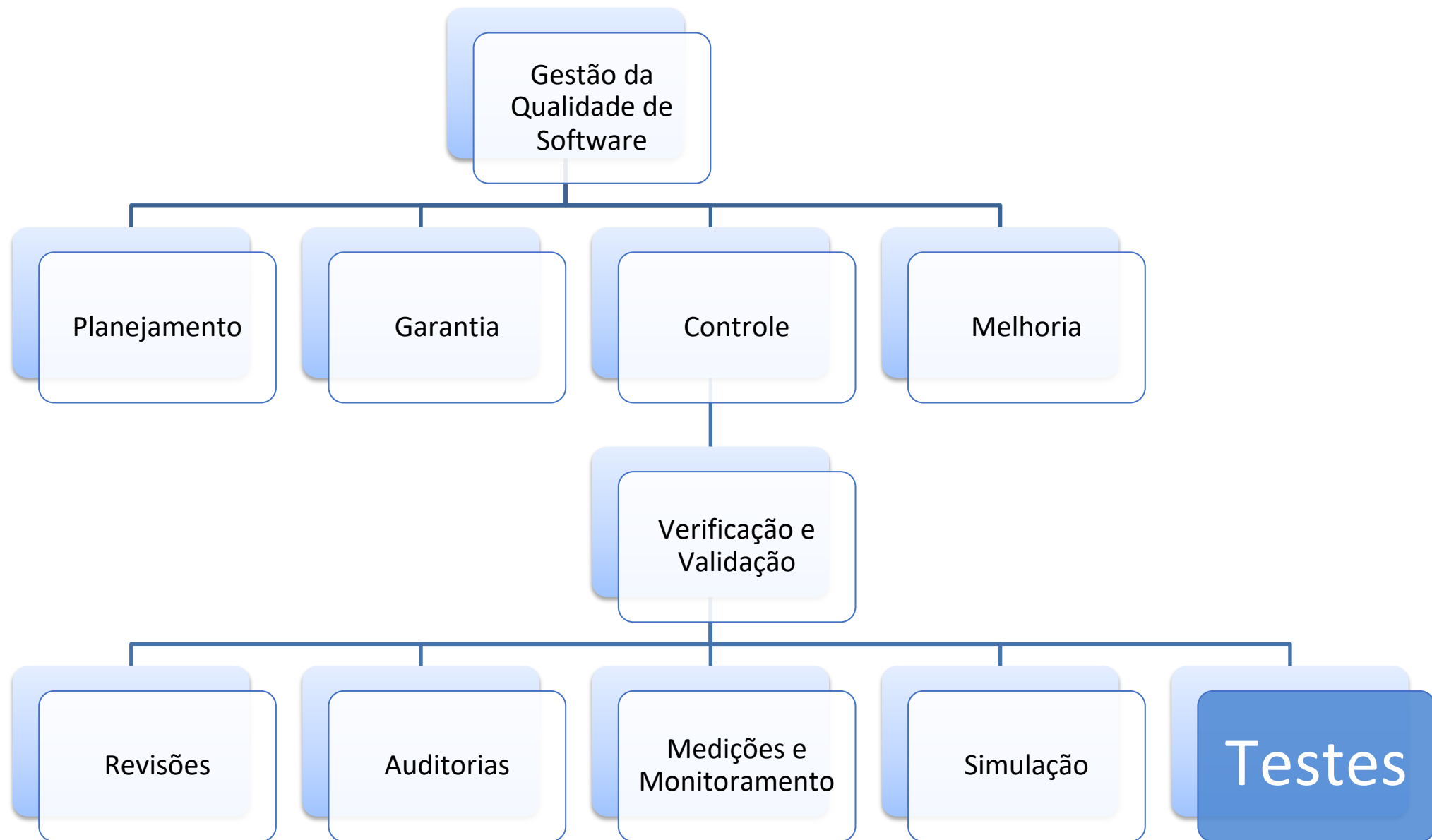
- **Verificação:**
 - Atividade para assegurar que software produzido está de acordo com suas especificações
 - “Nós construímos o software corretamente?”

Controle da Qualidade de Software

- **Validação:**
 - Atividade para assegurar que software produzido está de acordo com as expectativas do cliente
 - “Nós construímos o software correto?”

Controle da Qualidade de Software

- **V&V** abrangem um amplo conjunto de atividades:
 - Revisões
 - Auditoria
 - Medição e monitoramento
 - Simulação
 - **Testes**



Residência em TI Gerência de Configuração e Teste de Software

Prof. Eiji Adachi