

# ENGENHARIA DE SOFTWARE I

*Introdução à engenharia de software*



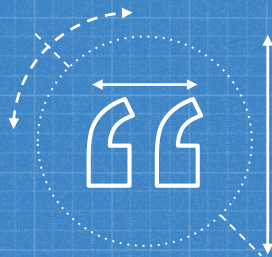
# O QUE NÃO É ENGENHARIA DE SOFTWARE?



Uma disciplina de programação.  
Uma disciplina de cálculo.



# O QUE É ENGENHARIA DE SOFTWARE?

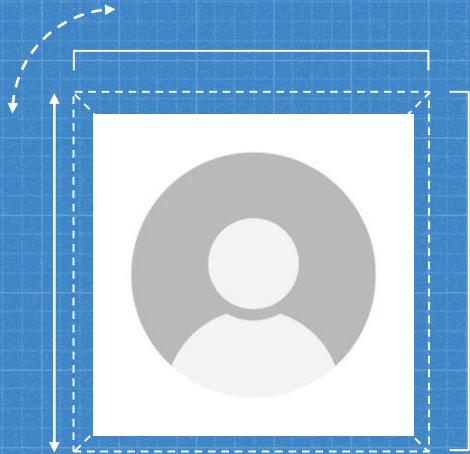


Área da computação destinada a identificar desafios e propor soluções que permitam desenvolver softwares de forma mais produtiva e de qualidade.



# Comentário de um dos participantes de uma das primeiras conferências de engenharia de software:

"O problema é que certas classes de sistemas estão colocando demandas sobre nós que estão além das nossas capacidades e das teorias e métodos de projeto que conhecemos no presente tempo. Em algumas aplicações não existe uma crise, como rotinas de ordenação e folhas de pagamento, por exemplo. Porém, estamos tendo dificuldades com grandes aplicações. Não podemos esperar que a produção de tais sistemas seja fácil."





# MAS... PROBLEMA EM QUÊ?

- Requisitos
- Comunicação
- Visão e organização
- Processos
- Produtividade
- Orçamento
- Complexidade do Código
- Alteração/Manutenção
- Etc.



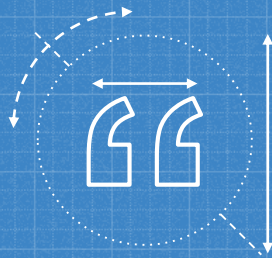
## DE MODO GERAL...

*Andar vendado em um labirinto!*





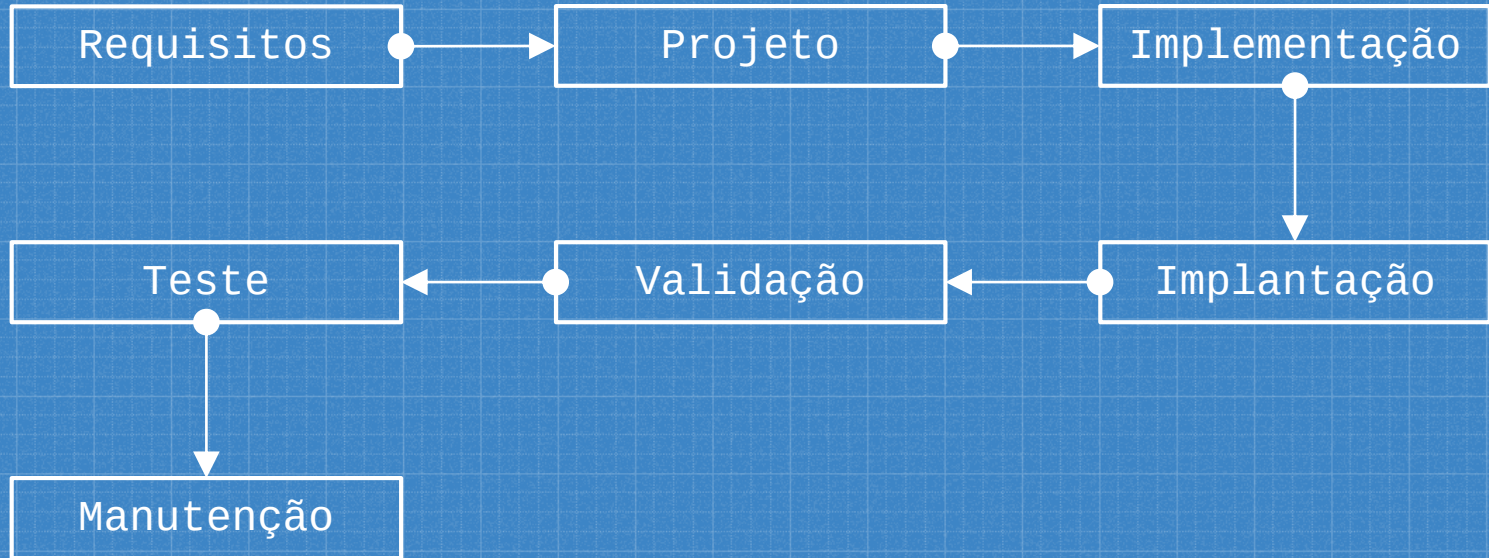
# CICLO DE VIDA DE SOFTWARE



**Estrutura básica que define quais etapas são importantes no desenvolvimento de um software.**

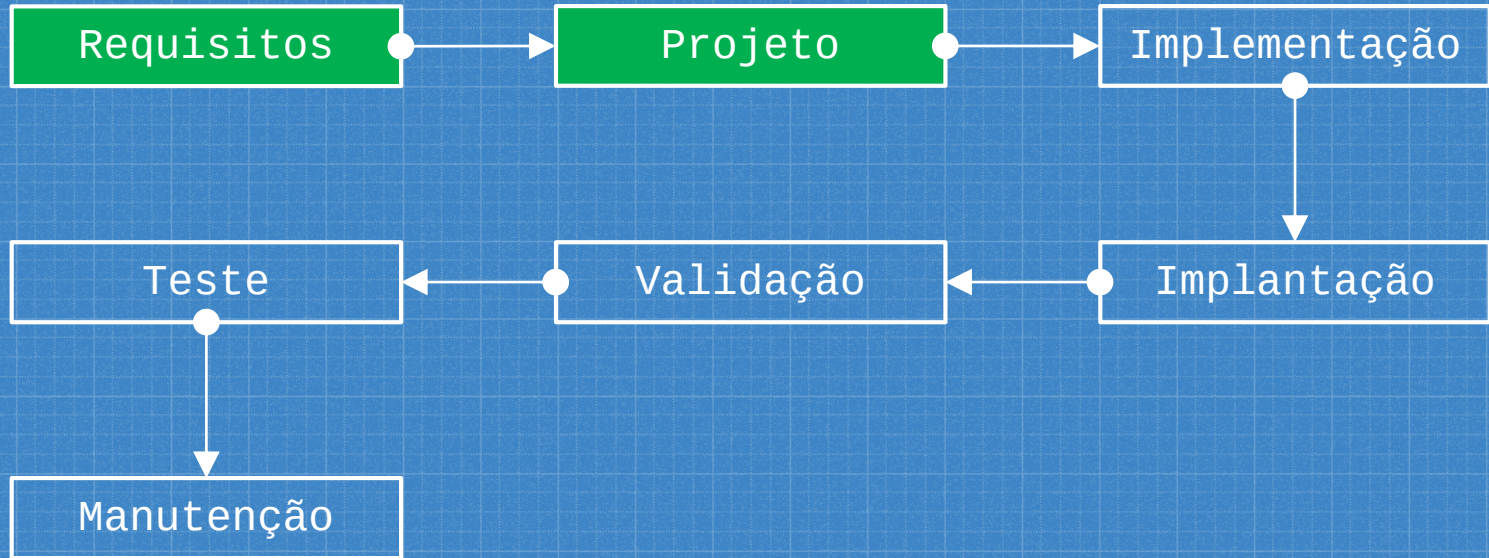


# CICLO DE VIDA DE SOFTWARE



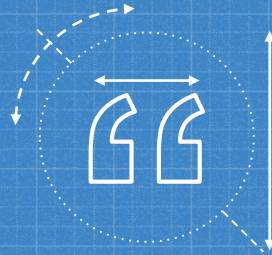


# CICLO DE VIDA DE SOFTWARE





# PROJETO DE DESENVOLVIMENTO DE SOFTWARE



**Estudo que define como as etapas do ciclo de vida de software devem ser realizadas.**



# PROJETO DE DESENVOLVIMENTO DE SOFTWARE

Existem muitas metodologias diferentes, mas que podem ser separadas em dois tipos:

- Processos em cascata
- Processos ágeis



# PROJETO DE DESENVOLVIMENTO DE SOFTWARE

Existem muitas metodologias diferentes, mas que podem ser separadas em dois tipos:

- Processos em cascata
- Processos ágeis

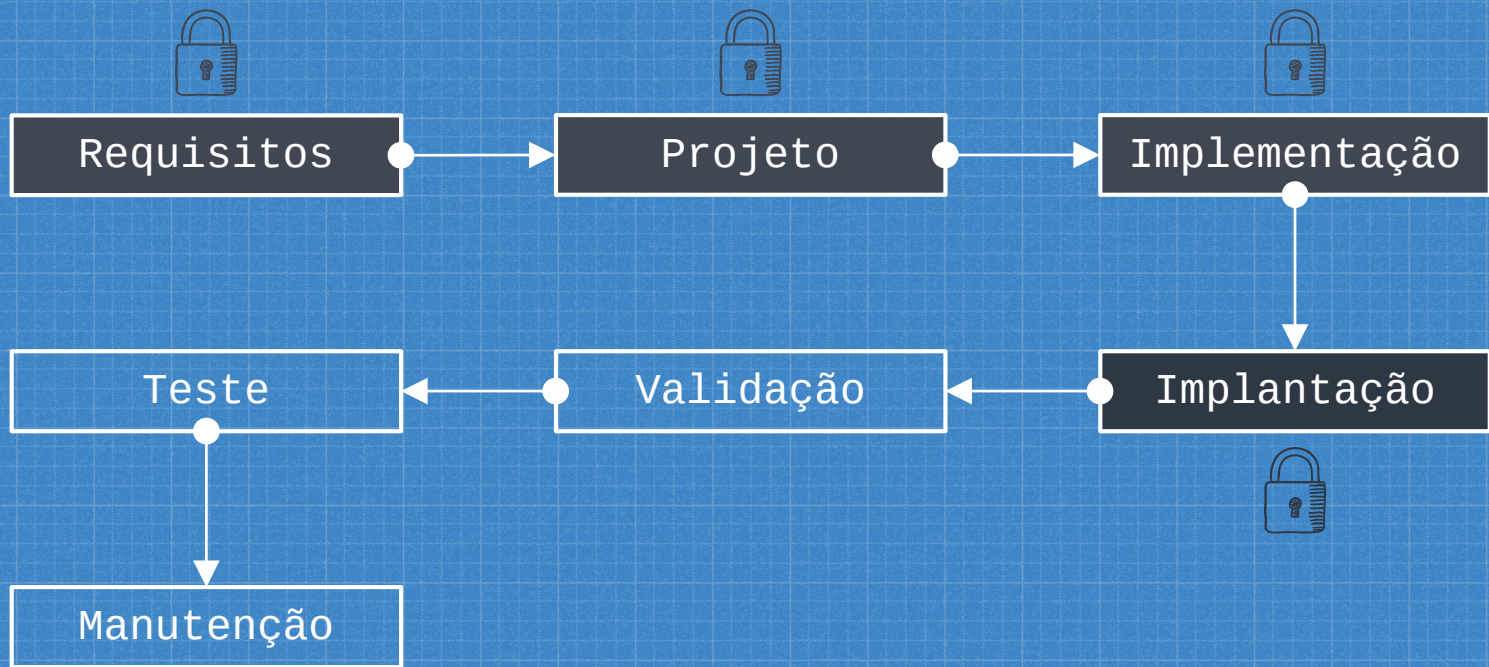


# PROCESSOS EM CASCATA

- Termina-se completamente uma etapa para começar outra.
- Exige que todos os pontos necessários para a criação do software sejam de conhecimento da equipe.
- Difícil adotar quando a ideia do software ainda é vaga.
- Em caso de ser inevitável voltar etapas, quantos mais distantes essas etapas estiverem, maior será o impacto negativo (estresse, retrabalho, custo, prazos).



# PROCESSOS EM CASCATA



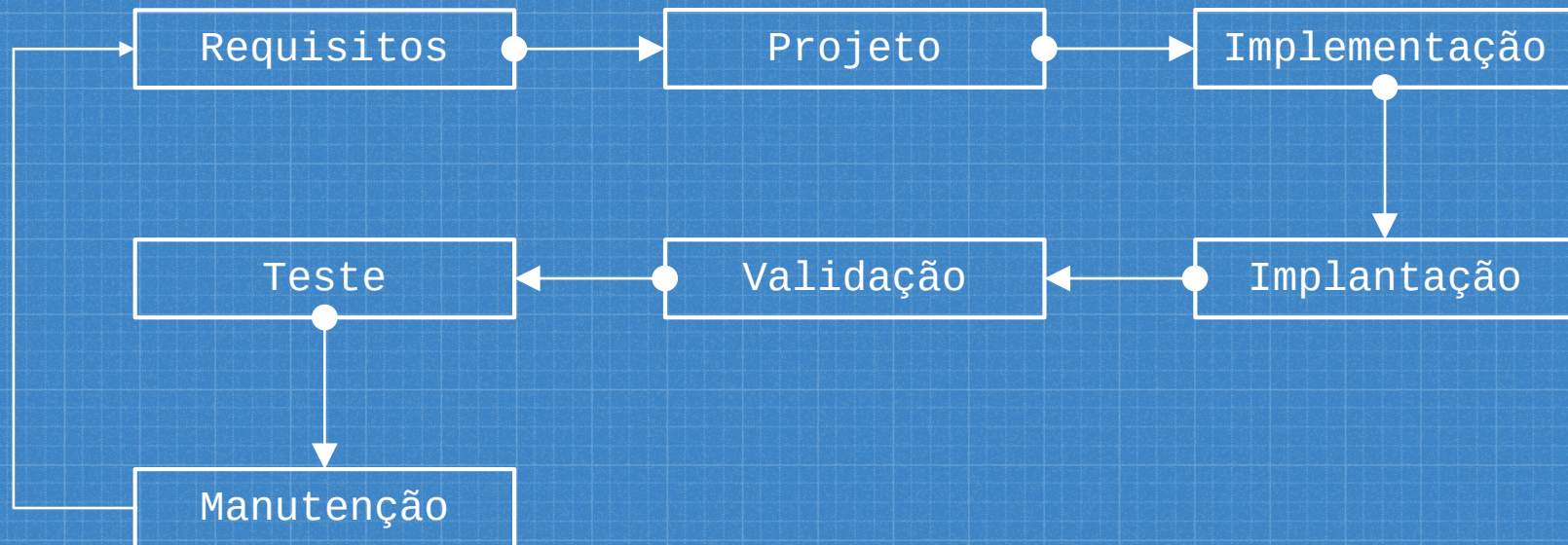


# PROCESSOS ÁGEIS

- As etapas podem ser revisitadas e algumas até mesmo feitas de forma simultânea (cada pessoa em uma etapa, é claro).
- Exige que uma pessoa ou grupo seja responsável por ter uma visão do todo, para guiar a equipe e resolver problemas.
- Grandes empresas utilizam esse tipo de processo.
- O custo de manutenção tende a ser mais barato, e os riscos amenizados.

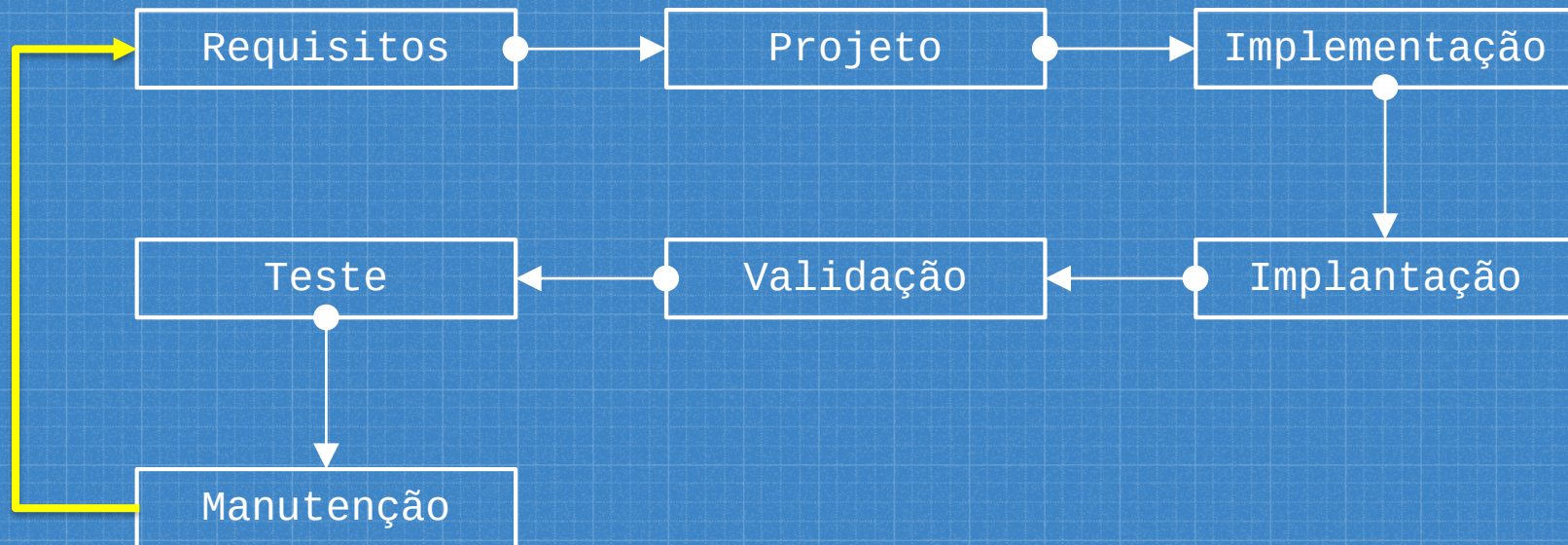


# PROCESSOS ÁGEIS





# PROCESSOS ÁGEIS



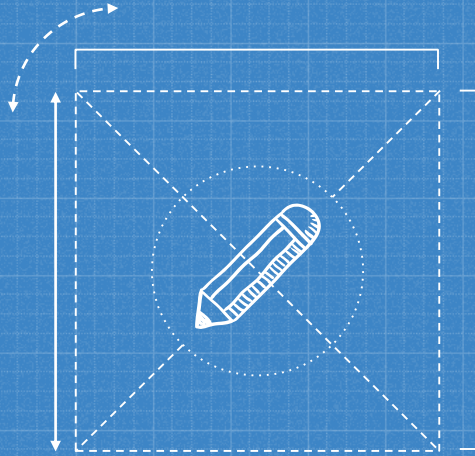


# PROCESSOS ÁGEIS



*Esse não é o único layout de processos ágeis.*





# SPOILER DO SEMINÁRIO

Cada aluno ou grupo será responsável por apresentar para o resto da sala um processo ágil, e depois teremos uma aula para discutirmos juntos todas as abordagens.