

Processo de Desenvolvimento de Software

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Departamento Acadêmico de Gestão e Tecnologia da Informação
Tecnologia em Análise e Desenvolvimento de Sistemas

Prof. Fellipe Aleixo (fellipe.aleixo@ifrn.edu.br)

Introdução

Processo de Desenvolvimento de Software ou
Software Development Process

- “Engenharia de Software é uma forma de engenharia que aplica os princípios da Ciência da Computação e matemática para alcançar soluções com o melhor custo-benefício para o problema do software”
 - *Software Engineering Institute* – SEI, 1990.
- “Engenharia de Software trata-se da aplicação sistemática, disciplinada, quantificável para o desenvolvimento, operação e manutenção de software”
 - IEEE, 1990.

Evolução do Software

- No início os softwares eram muito pequenos, dadas as limitações do hardware
- Com o crescimento do poder computacional, cresce também o tamanho e a complexidade do software
- Várias técnicas surgiram para ajudar na administração dessa complexidade
 - Técnicas ligadas à linguagens de programação
 - Aprofundamento dos estudos na engenharia de software
 - Arquitetura de software
 - Ferramentas CASE

A Crise do Software

- Inicialmente identificada nos **anos 60**, mas ainda hoje apresenta os seus efeitos
 - Um em quatro projetos de software falha na entrega
 - 20% de taxa de rotatividade de pessoal tida como normal
 - Grandes sistemas levando de 3 a 5 anos para serem desenvolvidos
 - Muitos deles se tornando obsoletos antes de serem entregues
 - Manutenção de software responsável pelo maior custo relacionado a computação para a maioria das empresas da área

Definição

- Um processo de desenvolvimento de software é a estrutura utilizada para o desenvolvimento de um produto de software
- Entre os seus sinônimos estão "ciclo de vida" e "processo de software"
- Há muitos modelos para esses processos, cada um descrevendo abordagens diferentes para uma variedade de tarefas e atividades a serem executadas durante o processo

Processo de Desenvolvimento de Software

- Investigar os requisitos dos usuários (**análise**)
- Claramente definir as características necessárias ao sistema (**especificação**)
- Criar (ou adaptar) uma solução adequada (**projeto**)
- Desenvolver a solução proposta (**implementação**)
- Garantir que a solução resolver o problema originalmente levantado (**teste**)
- Garantir que a solução funcione corretamente no seu contexto (**integração**)
- Modificar a solução de trabalho quando novos requisitos forem identificados (**manutenção**)

Análise

- Muitos (todos?) os usuários não sabem exatamente o que eles querem
 - Porque muitos não sabem exatamente o que eles fazem no dia a dia
- Portanto a análise requer que o desenvolvedor se torne intencionalmente especialista no domínio do usuário
- Fase 1: **Descoberta** (escutar e observar)
- Fase 2: **Refinamento** (interrogar e esclarecer)
- Fase 3: **Modelagem** (sugerir e verificar)
- Saída: entendimento suficiente do problema para escrever um documento de especificação de requisitos

Especificação

- Última fase da tarefa de análise
- Precisa escrever de **forma não ambígua** qual é o **comportamento requerido**
 - Notações formais
 - Documentos estruturados
 - Exemplos
- Saída: uma especificação dos requisitos que de forma não ambígua comunique ao projetista as características requeridas para o sistema

Projeto

- Desenvolver uma **solução que atenda aos requisitos**
 - Com base da experiência acumulada (e técnicas padronizadas)
- Geralmente precisam inovar em um certo nível
- Pode gerar várias possíveis soluções
 - Pode usar de alguma métrica para selecionar uma delas
- Saída: um documento de projeto que de forma não ambígua comunica o projeto para aqueles que iram implementar o mesmo

Implementação

- Escrever o código
- Documentar o código
- Reparar erros no código
- Preparar o código para ser testado
- Retornar informações ao projetista e/ou analistas
- Enviar informações ao testador e/ou integrador
- Saída: código de trabalho (e a documentação associada atualizada) preparado para ser testado

Teste e Integração

- Precisa **checar se a implementação corresponde ao projeto** (e esta funciona)
- Precisa **checar se a implementação atende aos requisitos** (e esta funciona corretamente)
- Deve testar os módulos individuais e o sistema por completo
- Então testa a interação com o(s) ambiente/software(s)/dados/etc. existente(s)
- Saída: código devidamente testado, funcionando corretamente

Manutenção

- Os requisitos dos usuários mudam com o tempo
- Mesmo teste exaustivos podem não descobrir todos os problemas antes da entrega do software
- Portanto, o software deve também mudar no decorrer do tempo
- Mudanças nos requisitos podem culminar em implementações e testes extras, ou trabalho adicional de projeto, ou até mesmo de análise

Atividade Transversal – Planejamento e Gestão

- Agendamento de tarefas em seus devidos momentos
- Alocação dos recursos necessários para que as tarefas tenham todas as condições de atingir os seus objetivos
- Acompanhamento das demais atividades, avaliação da sua eficiência, busca de formas de maximizar a mesma
- Responsabilidade de maximizar o ROI
- Responsabilidade de acordar com o cliente os prazos e características das entregas a serem realizadas

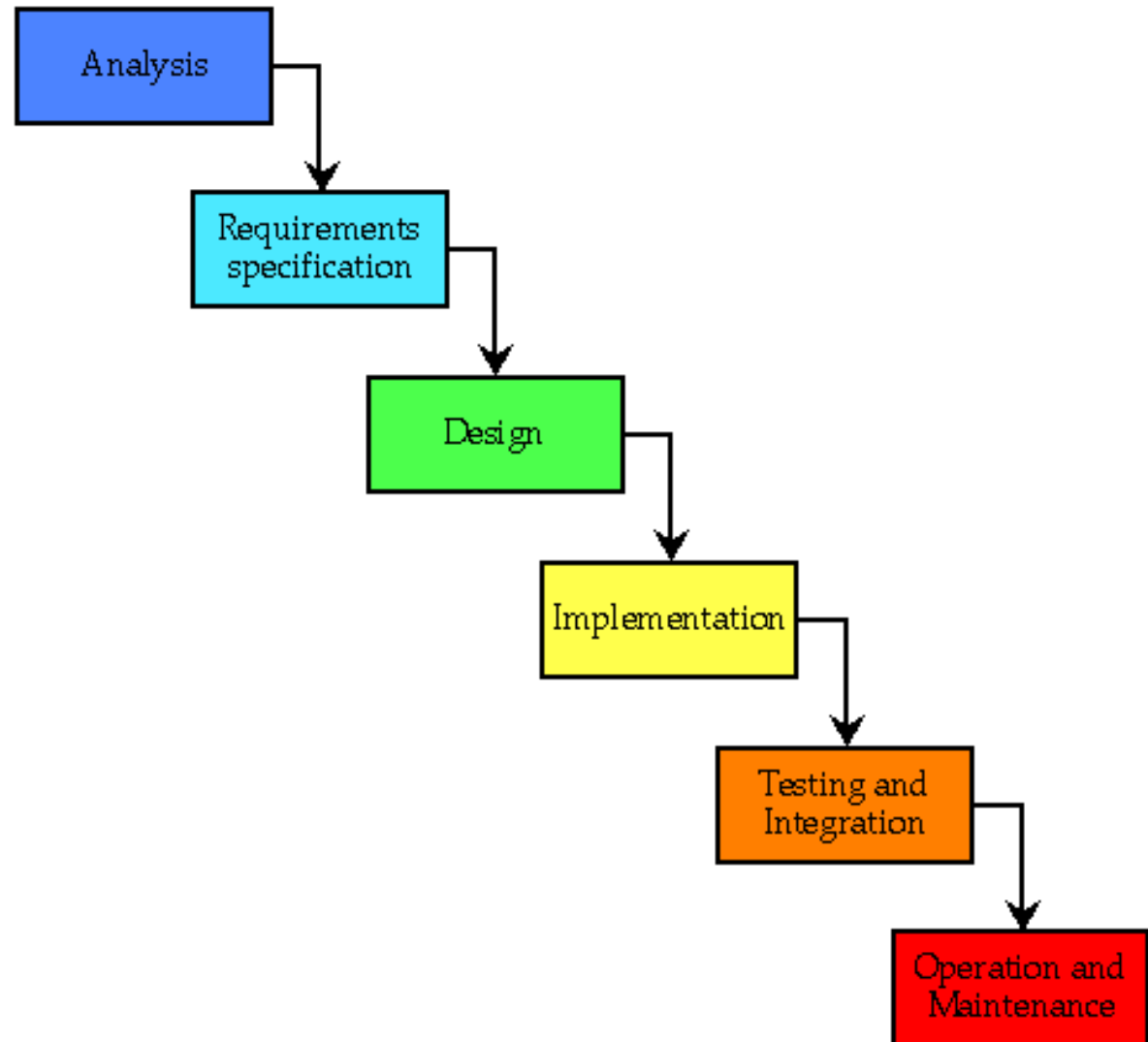
Modelos de Desenvolvimento de Software

Modelos de Desenvolvimento de Software

- Há um bom tempo, vem se tentando encontrar um processo ou metodologia **previsível** e **repetível** que melhore a produtividade e qualidade
- Vários modelos foram idealizados com o intuito de “organizar” o processo, podendo assim redundar em mais eficiência e menor custo para o mesmo

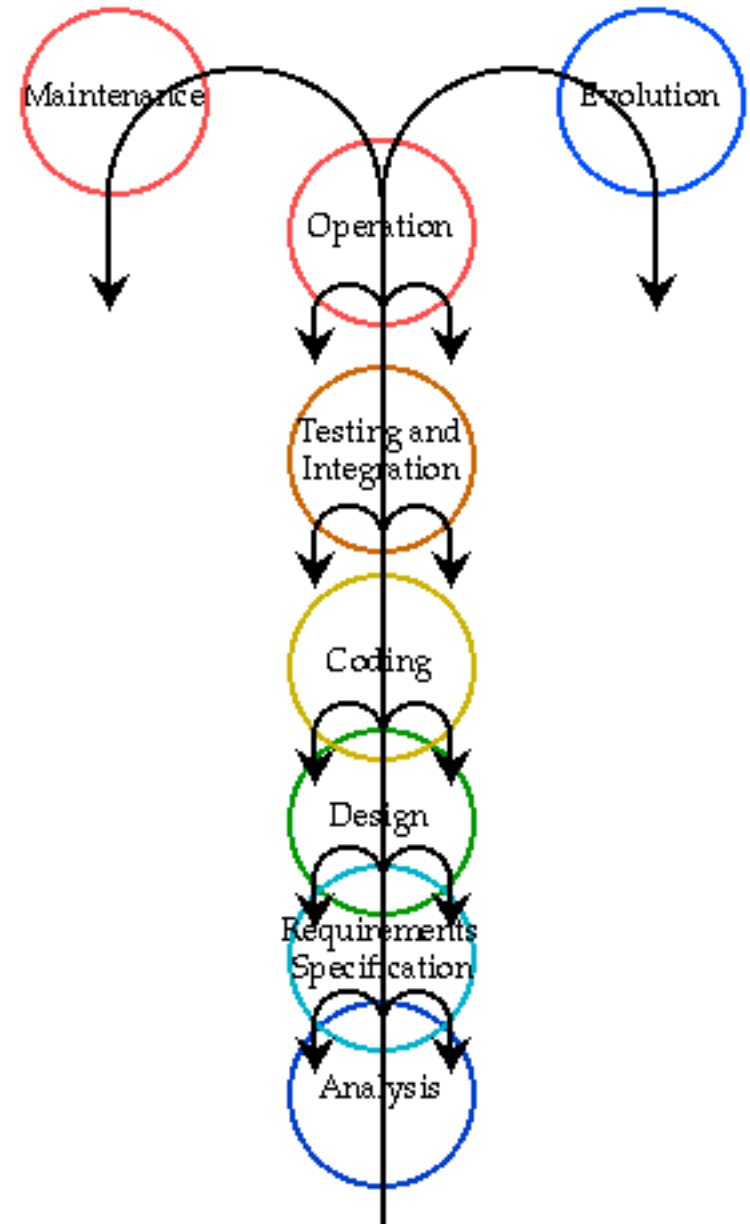
Modelo Tradicional em Cascata

- Modelo mais simples possível
- Fases executadas de forma seqüencial



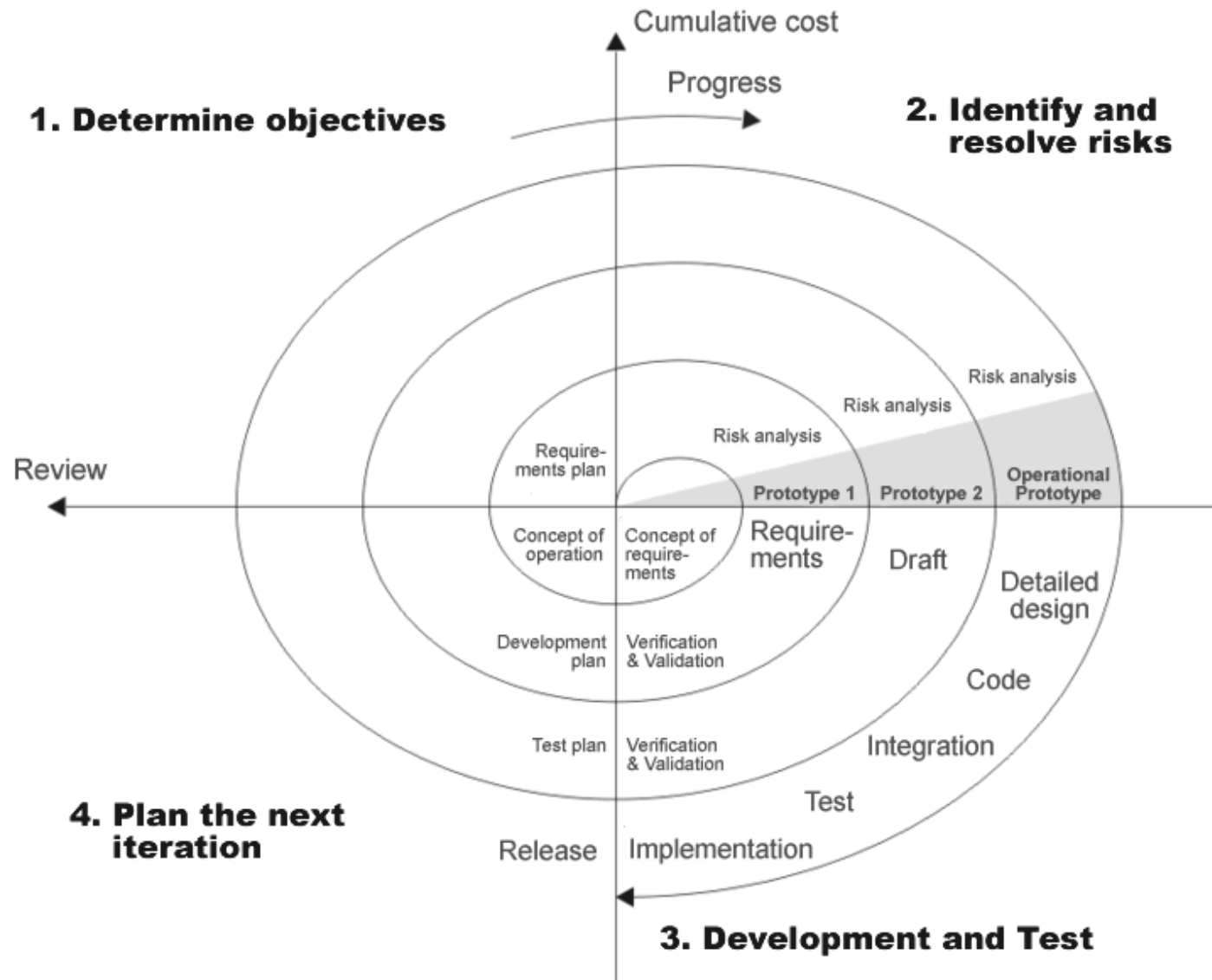
Modelo em Fonte

- Baseado no modelo de cascata
- Mas, observa que a sequência sempre contém ciclos
- Reflete o fato de que algumas fases não podem iniciar antes de outras
 - E que algumas fases são intercaladas



Modelo em Espiral

- Sugerido por Boehm em 1988
- Atividades que se repetem e geram um incremento



Desenvolvimento Baseado em Prototipação

- Variações no tema "construa alguma coisa e veja se é isso que é desejado"
- Pode compor um processo de desenvolvimento completo - Programação Exploratória
 - ou pode ser a simples antecipação do ciclo de projeto e implementação
 - ou pode ser parte de uma abordagem evolucionária

Desenvolvimento Iterativo e Incremental

- O desenvolvimento iterativo defende a construção inicial um pequeno pedaço do software, que vai crescendo gradativamente
 - Ajudando os envolvidos no processo a descobrir mais cedo possíveis problemas ou inconformidades, antes que possam levar ao desastre do projeto
- Processos iterativos são preferidos pelos desenvolvedores comerciais porque oferecem o potencial de atingir os objetivos de projeto para um cliente que não sabe como definir o que ele quer

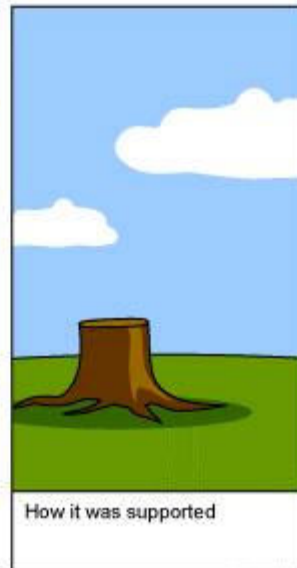
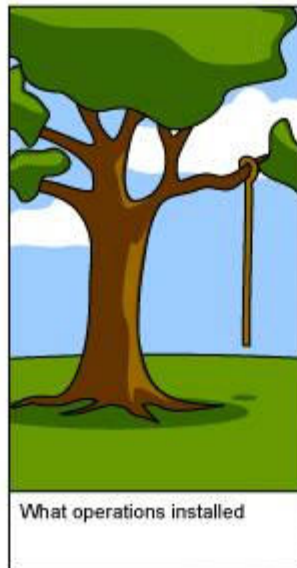
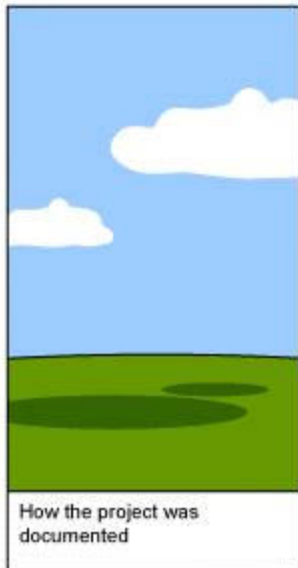
Desenvolvimento Ágil

- O desenvolvimento ágil de software defende alguns pontos de vista em detrimento de outros:
 - Indivíduos e interações sobre processos e ferramentas
 - Um software funcionando sobre uma documentação compreensível
 - Colaboração com o cliente sobre negociação de contratos
 - Resposta à mudança sobre seguir um planejamento
- Processos ágeis usam o “feedback”, ao invés do planejamento como o seu mecanismo de controle primário
 - O feedback é orientado por testes e releases periódicos do software envolvido

A Verdade Desagradável

- Cada um desses modelos é apenas teoria
 - uma simplificação para explicar o que realmente acontece
 - ou uma sugestão do que deve acontecer
- Eles são, no máximo, apenas aproximações da realidade
- Eles são baseados em pressupostos sobre os tipos de problemas que são comumente resolvidos, sobre as expectativas dos usuários, sobre os recursos disponíveis, sobre prazos, ferramentas, complexidade das tarefas, etc.
 - Nenhum desses pressupostos é 100% válido em todos os casos particulares
- Portanto não existe nenhuma "bala de prata" que sempre garanta que o desenvolvimento de grandes sistemas seja fácil

O que se quer evitar...

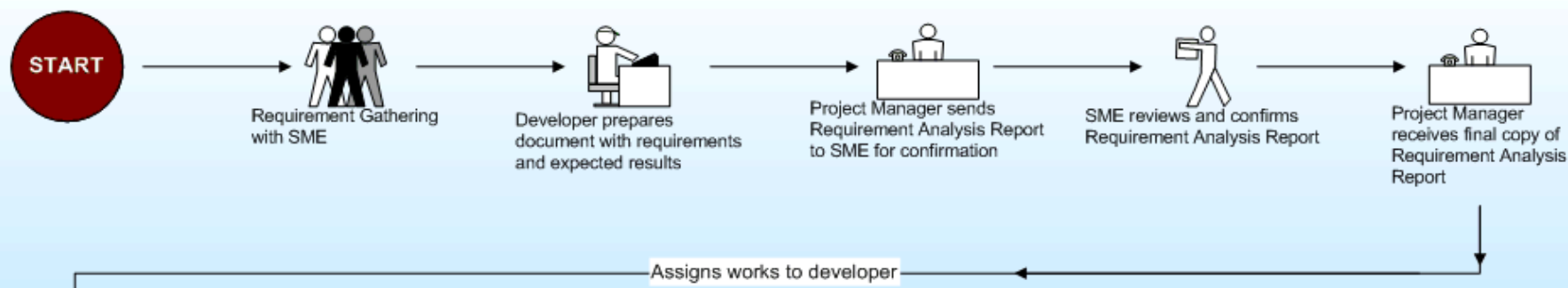


Um exemplo simplificado ilustrativo

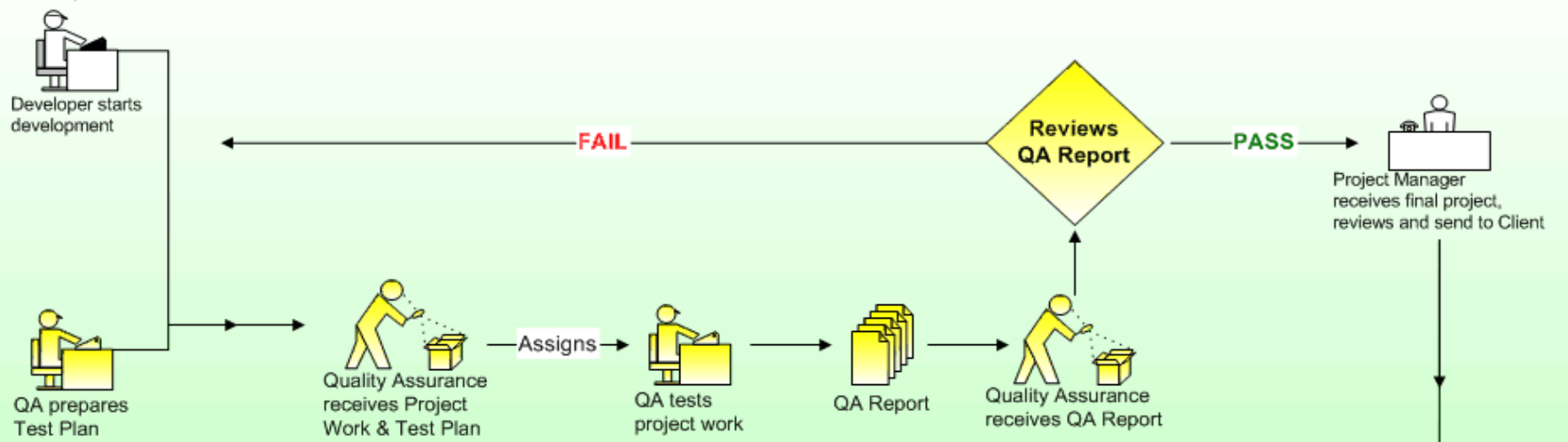
Ilustrado em três fases: requerimento, desenvolvimento e entrega

Software Development Process

REQUIREMENT PHASE



DEVELOPMENT PHASE



DELIVERY PHASE

