METODOLOGIAS ÁGEIS E CICLO DE DESENVOLVIMENTO DE SOFTWARE

Processo de software

É como se fosse um roteiro.

Não existe uma definição correta para as atividades de um processo.
Atividades principais podem variar, mas são convergentes.
Curso baseado no Pressman.

Exemplo:
Comunicação
Planejamento
Modelagem
Construção
Entrega

Fluxo de processo: Linear, Interativo, Evolucionário, Paralelo
Cada processo é construído de tarefas!

Padrões de processo
Certos templates de processo que resolvem determinados problemas

MODELO PRESCRITIVO

Ou tradicional, roteiro sequencial.

Modelo Cascata - Clássico

Modelo tradicional que exige bastante documentação
Sequencial - cada estágio inicia quando acaba o anterior

Interessante pq ele é linear e sistemático, tudo é bem definido.
Mas isso não é tão interessante quando os requisitos não são bem especificados. E o cliente nunca sabe tudo o que ele quer no início. E ele vai demorar muito para ver o projeto dele funcionando.

Modelo V

Ele é bem parecido com o Cascata, mas você vê melhor a relação com as atividades de teste.

Modelo de processo incremental

Existe uma sequência linear, mas vai entregando incrementos.
Pode-se utilizar prototipagem, caso o escopo não esteja tão definido.
Não é por ser incremental que ele deixa de ser Cascata.

Modelo evolucionário
Ideal para projetos grandes, extensos

Problemas da prototipação: acomodar com a qualidade global do projeto

Modelo concorrente
Estados de tarefas, as atividades podem acontecer de forma concorrente

MODELOS ESPECIALIZADOS - pq necessita a utilização de técnicas, dificilmente pode ser usado para qualquer coisa.

- vai de acordo com a necessidade do software

Modelos de métodos formais
Não é muito visto, pois é bem complexo. Tem que ter um embasamento formal muito grande.
Garante a confiabilidade do sistema, por exemplo, quando faz um sistema para avião (onde há risco de vida).
Utilizados para elaboração de sistemas computacionais dando prioridade a sua coesão, isto porque estes métodos são desenvolvidos a partir de princípios matemáticos que garantem a sua exatidão na capacidade de expressão das ideias vinculadas ao projeto de software.

Modelo orientado a aspectos
Pontos de interesse da aplicação.
Separação de responsabilidades, considerando funcionalidades que são essenciais para um grupo de objetos, mas não são de responsabilidade direta deles.

PROCESSO UNIFICADO - mostra transição e necessidade de mudar como se vê os modelos tradicionais
Ser iterativo e incremental satisfaz mais o cliente.
Concepção
Elaboração
Construção
Transição - fase onde pega bugs, processo de transição final. Recebe feedback do usuário final
Produção

MANIFESTO ÁGIL

2001 - criado um documento Manisfesto Ágil - 17 homens (ex: Martin Fowler)
São 12 princípios
Quebrar o paradigma do prescritivo.

- Custo menor
- Não é preciso estimar longos prazos
- Imprevisibilidade

Extreme Programming XP

Iniciada por Kent Beck
Mais indicado para planejamento de software

Originado nos anos 80, pensado com o paradigma orientado a objetos.
Focado diretamente na produção do código.
Planejamento
Planning Poker - ouvir de quem vai desenvolver
Existem ciclos que geram incrementos para o software
Projeto
Atributos, métodos, classes.
Utiliza a prototipação para mostrar ao cliente
Codificação
TDD
2 pessoas sentam juntas para desenvolver
Testes

Scrum

Esse nome vem de uma ação do rugby - equipe unida para o sucesso.
Pode servir para qualquer tipo de projeto;

- 3 pilares do Scrum: Transparência, Adaptação, Inspeção

Processo: Ideia - User Stories - Product Backlog - Planning meeting
Sprint leva de 2 a 4 semanas

Scrum não existe hierarquia (ex: gerente de projeto). Decisões relacionadas ao produto são decididas pelo time todo.

Product Owner
Diz o que o produto deve ter; Backlog do produto;
Desenvolvedores

Qualquer um da equipe que tenho o objetivo de criar o produto;
Preocupados com o backlog da sprint
Scrum Master
Fazer intermediação necessária para que a tecnologia funcione; Mantém as pessoas sempre informadas de suas tarefas

Eventos
Sprint - manter garantia e qualidade do produto
Planning - O que vai ser feito e como será feito
Daily - 15 a 20 minutos, entender se a meta da sprint está sendo alcançada
Sprint Review - Mais focado na entrega
Sprint Retrospective - Fechamento da sprint, como melhorar sempre

Outros modelos ágeis

DSDM
Tem restrição específica de tempo; Entrega somente o *suficiente* em menos tempo;
As funcionalidades se tornam variáveis e o tempo é fixado;

- Existem papéis bem definidos nessa metodologia

Processo Unificado Ágil

Kanban
Surgiu no Japão - não é direcionado a desenvolvimento de software
É muito voltado ao aspecto visual;

TESTES NO MUNDO ÁGIL

O teste existe sempre
A nível de código continua existindo o teste de unidade…

- Ter olhar crítico para saber o que testar ou não
- São várias atividades implícitas,

Diferenças nas abordagens ágeis
No Scrum, os testes já começam na planning.
Todo mundo do time sabe da validação e verificação
Teste exploratório tem tudo a ver com modelo ágil;

Produtos de trabalho:
Ir direto ao ponto, analisar os riscos envolvidos

Nível de teste no modelo ágil
Foco no modelo ágil: teste unidade e teste de aceite

Status de teste
Atualizar a equipe diariamente;

MÉTODOS DE TESTE

TDD - vários ciclos - garante que o código sempre estará funcionando
Desenvolve o esqueleto do teste primeiro, depois escreve o código

ATDD - orientado por teste de aceite - não exclui o TDD

BDD - orientado a comportamento - utilizado em conjunto com o TDD

DEV OPS - é uma cultura, uma metodologia e se integra muito bem com a agilidade

Diferença AOP (orientada a aspectos) e POO (orientada a objetos)
Os paradigmas de programação diferem na forma como cada elemento dos programas é representado e como cada etapa é definida para solucionar problemas. Como o nome sugere, o OOP se concentra em representar problemas usando objetos do mundo real e seu comportamento, enquanto o AOP lida com a divisão dos programas para separar preocupações transversais

Um processo iterativo é aquele que faz progresso através de tentativas sucessivas de refinamento. Por exemplo, uma equipe de desenvolvimento faz sua primeira tentativa para construção de um software, porém, existem pontos de informação falhos ou incompletos em algumas partes