

## **Pesquisa Introdução a Métodos Ágeis**

1. **Especificação:** Etapa onde é definido as funcionalidades do software e suas restrições sobre operações. O sistema deve satisfazer as necessidades do cliente, atendendo a todos os requisitos. Existem algumas etapas nesse processo:

Estudo de viabilidade: É feito uma estimativa da viabilidade, considerando restrições, a tecnologia usada, orçamento, prazo, etc.

Elicitação: Onde é levantado os requisitos do sistema, através de entrevistas. O que são os requisitos; R: Ação que deve ser executada pelo sistema

Especificação de requisitos: Onde é colocado em documento, catalogado e classificado todos os requisitos obtidos através das técnicas e entrevistas feitas.

Validação dos requisitos: O documento é avaliado de uma forma geral, quanto sua consistência, completude e ao realismo. O software também é testado várias vezes até chegar ao resultado esperado.

**Projeto e implementação:** Esse processo inclui pelo menos três fases principais.

Projeto Arquitetural: Fase onde será definido a estrutura do software, as suas interfaces e as estruturas de dados que serão utilizadas no sistema.

Projeto Detalhado: Fase onde é definido uma solução detalhada para cada módulo do projeto preliminar.

Implementação: Fase onde é transcrito todas as decisões do projeto e requisitos do documento para uma determinada linguagem de programação.

Evolução: O custo para manutenção de softwares geralmente é maior que o seu custo de desenvolvimento, assim menos cada vez mais os sistemas tendem a ser desenvolvidos/adaptados a partir d outros sistemas, não tendo o desenvolvimento do zero.

2. **Prototipação:** A etapa de prototipação pode reduzir os custos do desenvolvimentos assim também como evitar prejuízos, nesta etapa a equipe deve:
  1. Estabelecer objetivos do protótipo (Criando um plano de prototipação)
  2. Definir a funcionalidade do protótipo (Fazendo as definições gerais)
  3. Desenvolver protótipo (Criar um protótipo executável para ser testado)
  4. Avaliar o protótipo (Criar um relatório de avaliação)

Entrega incremental: O modelo de processo incremental aplica sequências lineares de forma escalonada, à medida que o tempo for avançando. Cada uma das sequências lineares gera um incremento do software, ou seja, uma nova funcionalidade. Podemos exemplificar com um sistema de cadastro de pacientes e marcação de consultas, na primeira versão do software entrega-se apenas o sistema com o cadastro de clientes, enquanto vão cadastrando todos os clientes os desenvolvedores vão criando a nova função para agendar consultas, entregando posteriormente uma nova versão do software com um incremento a mais. O modelo de processo incremental entrega uma versão de software a cada incremento, ou seja, um produto sem erros e pronto para o usuário utilizar. Mesmo que os primeiros incrementos sejam partes do produto, essas partes são operacionais e funcionam sem as outras.

### 3. Vantagens:

- Tem uma forte base e uma boa comunicação entre desenvolvedores e usuários.
- Existe um grande envolvimento do usuário e do cliente. Isto às vezes acaba levando a mal entendimentos por causa das divergências entre os desenvolvedores e usuários que rapidamente podem ser superados. Porque há resultados mais rápidos e “tangíveis”, os usuários também serão capazes de dar um melhor feedback do sistema.
- Os resultados mostrados a cada etapa e evolução do processo permitirão que os usuários tenham confiança em um bom resultado.
- A cada ciclo do processo do sistema os usuários e clientes poderão utilizar o sistema diretamente, sendo assim os “testadores” no processo de desenvolvimento, assim interagindo com o desenvolvimento do sistema em tempo real.
- Os riscos podem ser melhor administrados por estar lidando com entrega do software por etapas.
- Os sistemas mais complexos passam a ser mais claros e menos complexos, pois o cliente tem acesso ao projeto desde o início, não sendo necessário um treinamento do zero.
- Ao fim de cada ciclo pode-se ter um feedback para ter noção de como está o projeto, mesmo se ainda não estiver de acordo como o desejado, ainda há tempo para mudanças.
- Alterações nos requisitos podem ser rapidamente incorporadas no processo de desenvolvimento.

### Desvantagens:

- Cada fase de uma iteração é rígida e não se sobrepõem uns aos outros.
- Podem surgir problemas relativos quando à arquitetura do sistema, porque nem todos os requisitos estarão reunidos na frente de todo o ciclo de vida do software.

- Precisa ser relativamente pequeno, pois o projeto pode demorar muito tempo para ser concluído.

4. É um conjunto de metodologias utilizadas no desenvolvimento de um software. Para assim tornar mais ágil o processo de desenvolvimento do software, através de frameworks de processos para desenvolvimento de software, entre vários outros métodos, a maioria dos métodos visa minimizar os riscos pelo desenvolvimento do software em curtos períodos.
5. Satisfazer o cliente através da entrega adiantada e contínua do software. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, dando assim competitividade ao cliente. Entregar o software funcionando com frequência e, na escala de semanas até meses, em curtos períodos de tempos. As Pessoas relacionadas à negócios e desenvolvedores, devem trabalhar juntos durante todo o projeto, para que se tenha o resultado esperado. Método mais eficiente para transmitir informações a equipe de desenvolvimento.
6. Garantir a satisfação do consumidor, entregando rapidamente e continuamente softwares funcionais. Até mesmo mudanças tardias de escopo do projeto são bem vindas para garantir a vantagem competitiva. Softwares funcionais são entregues frequentemente, em semanas ao invés de meses. Cooperação diária entre as pessoas que entendem do negócio e os desenvolvedores. Desenvolvimento sustentável é um dos fatores de princípio para a abordagem desta técnica. O Design do software deve prezar pela excelência técnica. A simplicidade do processo torna-se mais atrativo e é essencial para o todo. As melhores arquiteturas, requisitos e projetos emergem de equipes organizadas que estão em convívio diariamente, tanto com clientes quanto com desenvolvedores.
7. Poderíamos citar várias dificuldades, entre elas: Apoio das instâncias superiores, Gerenciamento de equipes, Problemas técnicos, Interação com outros departamentos, Interação com clientes, e as pessoas, são as vezes o principal problema, seja por falta de comunicação como por falta de entendimento.