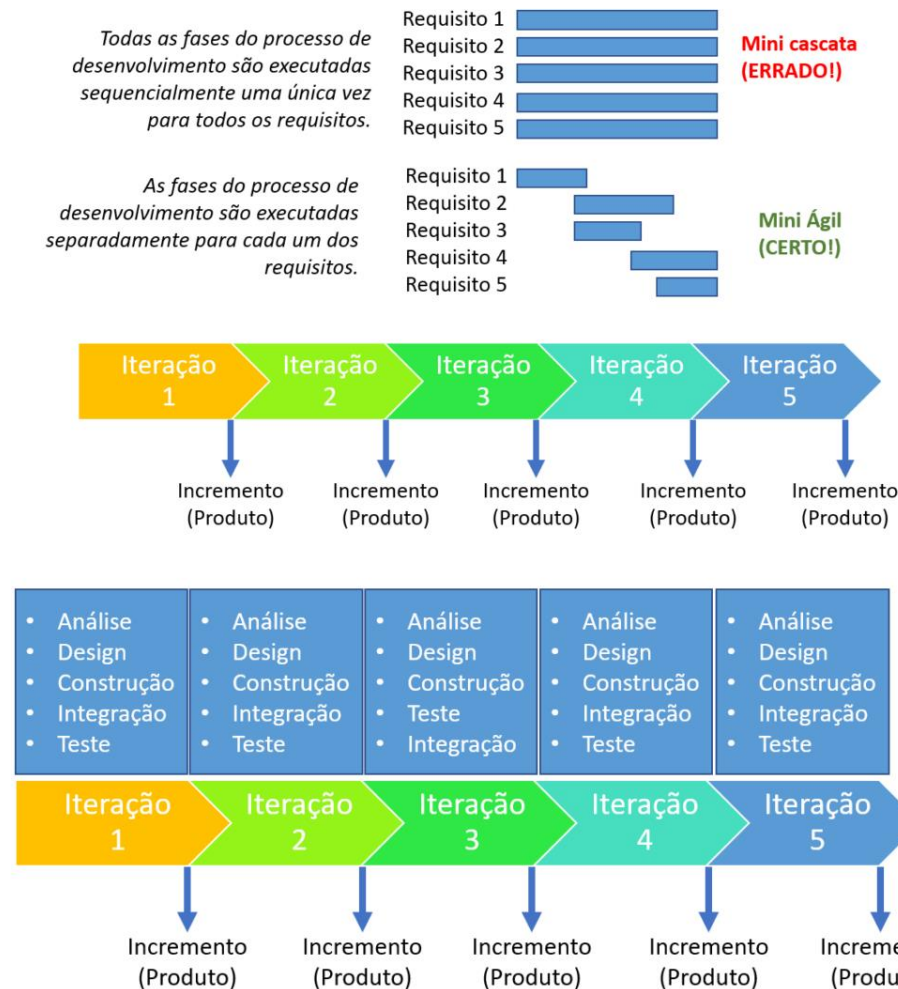


Método Ágil:



Cada iteração (ou Sprint) entrega uma parte utilizável do software para o cliente e assim ele vai vendo se o que ele tinha em mente é exatamente o que está sendo construído.

Este modelo é o que chamamos de iterativo e incremental, ou seja, a cada nova iteração entregamos um novo incremento de software.

Mas veja que isto não é a adoção de mini-cascatas em cada iteração! De forma alguma. O que se tem na verdade é a divisão dos requisitos em partes menores e a entrega dessas partes sendo feitas a cada nova iteração.

Assim, nos métodos ágeis as iterações não são divididas em fases. Por exemplo: não existe a fase Análise de Requisitos onde todos os requisitos daquela Sprint são definidos. O que se faz é selecionar um requisito (ou uma parte menor dele), realizar a análise, design, construção, integração e teste.

Nesse meio tempo, outro requisito pode ter sido escolhido e o ciclo (análise, design, construção, integração e teste) é executado para ele também. E assim se prossegue até o término da iteração.

É como se construíssemos, em uma obra de construção de uma casa, uma suíte por vez, ou seja, indicamos o que queremos que tenha nessa suíte, aí é elaborada a planta somente dessa suíte, depois ela é construída, testada e entregue.

Na próxima iteração, passaremos à construção da segunda suíte e assim por diante. Aqui já deu para perceber que os métodos ágeis não são a melhor forma de construir casas, não é mesmo? Mas sem dúvida, é o melhor método para construção de projetos que tenham escopo volátil e complexo de ser definido logo no início.

Resumindo Em resumo, os modelos ágeis ou iterativos vêm para resolver 3 falhas importantes dos modelos tradicionais: Os projetos reais construídos em alguns tipos de indústria (como a de software ou de produtos inovadores) raramente seguem o fluxo sequencial que o modelo prega.

É muito raro e difícil para o cliente saber e estabelecer explicitamente todas as suas necessidades. O modelo cascata é fortemente baseado nisso e tem dificuldade para adequar a incerteza natural que existe no início dos projetos. O cliente precisa ter muita paciência, o que raramente acontece.

Uma versão operacional (pronta para ser executada no cliente) não estará disponível até estarmos próximo ao final do projeto.

Se tivermos um erro grave nas etapas iniciais, como uma especificação mal compreendida e mal especificada, podemos ter um resultado desastroso, ou seja, o cliente se recusará a aceitar e meses de trabalho podem ter que ser jogados fora.