

Em métodos tradicionais de desenvolvimento de software (Ex: waterfall e v-model) utilizava-se uma abordagem sequencial que consistia em várias fases (Requisitos, Análise, Desenho, Implementação) em que estas eram percorridas uma única vez, sequencialmente (sem voltar atrás), o que pressupunha não só que os requisitos eram completos e estáveis, mas também que os primeiros protótipos do software apenas se encontravam disponíveis para apresentação aos stakeholders numa fase avançada do projeto, e que, por essa razão, os maiores riscos apenas eram controlados nesta altura.

O facto de utilizarmos esta abordagem sequencial tem como consequência um custo elevado para qualquer alteração significativa no software, principalmente modificações estabelecidas durante as fases iniciais do desenvolvimento (Requisitos / Análise), e que tende a crescer à medida que o software é implementado. Estas consequências levam a que esta abordagem evite qualquer tentativa de adaptação do software em desenvolvimento e, consequentemente, custos altos.

O que é inevitável ocorrer com estes modelos é que, devido à constante oscilação das necessidades do negócio, são exigidas adaptações nos requisitos e desenhos na qual assentava toda a implementação do software que, consequentemente, acaba por ter um conjunto de alterações muito significativas que, se não forem aplicadas, levam à desvalorização do negócio.

Os métodos ágeis foram desenvolvidos com um paradigma totalmente diferente dos processos tradicionais de desenvolvimento de software, tendo como principal objetivo a redução do custo da mudança ao longo do projeto, respondendo assim às adaptações previamente referidas.

Para atingir estes objetivos, os métodos ágeis defendem a utilização de uma abordagem iterativa/evolutiva. Esta abordagem defende que o software deve ser implementado através de vários incrementos, uma pequena parte do software no qual são selecionados um conjunto de requisitos e são rapidamente desenhados, implementados e testados. Os incrementos depois de implementados vão ser integrados e novamente testados (testes de integração), para então serem adicionados ao software em desenvolvimento. Todos os incrementos depois de integrados passam por um processo de feedback, com o objetivo de melhorar os próximos ciclos de desenvolvimento.

Esta metodologia não só permite várias versões do software implementadas cedo no processo de desenvolvimento, o que permite feedback por parte dos stakeholders e o controlo de riscos cedo, como também facilita e reduz muito significativamente o custo das adaptações e modificações durante o desenvolvimento devido ao seu paradigma (redução do custo da mudança ao longo do projeto).