

Modelo de ciclo de vida incremental para o desenvolvimento de software

Lucas Oliveira de Souza¹, Igor Schneider¹, Tiago Rios da Rocha^{1*}

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) -
Campus Ibirubá. Ibirubá, RS, Brasil.

A Engenharia de Software surgiu devido aos problema recorrentes no desenvolvimento de software, a partir da dificuldade de definir os requisitos, estimar prazos e custos corretamente. Para tentar resolver isso, utilizam-se modelos de ciclo de vida, de modo a organizar o processo de desenvolvimento para torná-lo mais eficiente. Todos os modelos possuem ao menos quatro fases essenciais em comum: análise, projeto, implementação e testes. Hoje em dia, há uma variedade de modelos que encaixam-se em diferentes estilos de projeto, propondo relações variadas entre as fases, conhecidos como Modelo de Ciclo de Vida. Porém, durante muito tempo, o modelo cascata foi o único aceito. Nesse modelo, o desenvolvimento do projeto percorre sequencialmente fases bem definidas, iniciando uma a partir da conclusão da fase anterior (Sommerville, 2011). Sugerido por Barry Boehm, o modelo incremental é uma tentativa de superar limitações do modelo cascata, como a impossibilidade de colocar o cliente em contato com o software em estágios iniciais do desenvolvimento. Seu funcionamento ocorre do seguinte modo: é desenvolvida uma versão inicial, ou núcleo, com todas as funcionalidades básicas do produto; em cima do núcleo, é então realizada uma série de incrementos até que a versão final seja alcançada; cada incremento é desenvolvido através de um ciclo de vida cascata completo (Sommerville, 2011). Por ser uma composição de modelos cascata, possui muitas das características desse modelo, como a dificuldade de prever o futuro do projeto e inflexibilidade da arquitetura, além de ser mais benéfico em projetos pequenos. Seu principal avanço em relação ao modelo cascata é a possibilidade de fragmentar o desenvolvimento em partes menores, permitindo um maior número de avaliações, o que reduz as chances de o produto final estar muito distante dos requisitos. Além disso, é fácil descartar o incremento mais recente em caso de equívoco, uma vez que o conjunto anterior como um todo é independente dele. Porém, é importante avaliar os riscos dessa abordagem, pois, devido ao número indefinido de incrementos que o software possa necessitar, há a possibilidade de que todo o orçamento seja consumido antes do término do projeto. O modelo incremental é simples e apresenta vantagens em relação ao cascata, entretanto, em projetos maiores, talvez seja mais vantajoso explorar outros modelos de ciclo de vida, mais adequados ao porte da aplicação.

Palavras-chave: Engenharia de software. Modelo de ciclo de vida. Modelo incremental.