

Babel digital

Organizações estão continuamente envolvidas no desenvolvimento de *software* para fazer frente às suas mais diversas necessidades: da gestão de materiais ao atendimento ao cliente. Diante de tais projetos, seus profissionais podem se ver em um mar de siglas. E o mundo digital é repleto delas, algumas impenetráveis. Sua proliferação ocorre tanto por razões técnicas como por razões comerciais. Alguns vêem o fenômeno como exemplo do comportamento de gueto dos profissionais de informática, pouco interessados nas reais necessidades e sempre prontos a lançar uma cortina de fumaça sobre suas atividades. Teorias conspiratórias à parte, vamos tomar três siglas que aparecem com frequência quando o tema é projeto de *software*: SW-CMM, UML e XP.

SW-CMM significa “Software Capability Maturity Model”, um modelo que descreve as chaves para processos eficientes de desenvolvimento de *software*.

UML significa “Unified Modeling Language”. Trata-se de uma tentativa de padronização das notações para as diversas fases do projeto de *software*: modelos semânticos, notação sintática e diagramas, permitindo um projeto visual do *software* bem antes de implementá-lo.

XP significa “Extreme Programming” – método que privilegia a comunicação informal entre os projetistas, a simplicidade de projeto, a revisão do código e testes constantes. Seu uso é destinado a equipes que estejam desenvolvendo *software* com requisitos imprecisos ou mutáveis.

Essas siglas podem ser utilizadas de forma complementar. Tomemos o caso do XP, que às vezes é considerado antagônico aos conceitos do SW-CMM e da UML. Ele é criticado como sendo uma forma indisciplinada de desenvolvimento de *software*, visto que nega a seqüência tradicional de projeto. Os defensores do XP afirmam que o processo de desenvolvimento de *software* se assemelha mais à redação de um livro – repleta de idas e vindas – do que à construção de uma casa – que começa por projeto arquitetônico e firmes alicerces. O XP contraria claramente o conceito de “maturidade” dos processos preconizado pelo CMM. Porém, para pequenas e médias equipes, várias características do XP podem se adaptar aos requisitos dos processos do CMM.

Diz-se que UML inviabiliza a adoção do XP. Entretanto, XP pode sim utilizar conceitos da UML, mas em uma fase posterior, a da documentação do projeto realizado.

Desta forma, CMM, UML e XP não devem ser considerados como alternativas excludentes; devem ser vistos pelas empresas e pelas equipes de desenvolvimento de *software* de acordo com o contexto de cada projeto.

Embora os defensores de um ou outro modelo eventualmente assumam “zelo religioso” em sua defesa, uma postura mais recomendável a esses profissionais, e principalmente a seus clientes, é uma postura pragmática, que permita a correta combinação de abordagens em benefício do produto final.



Nelson Lerner Barth
FGV-EAESP