

# A importância do Build e Release no DevOps



Ao analisar os projetos de software pelo Brasil é muito comum encontrar problemas relacionados a compilação (Build) e Release (Publicação) das aplicações. Os cenários são bem variados desde compilação de uma versão localmente e não realização do código fonte para o repositório, dependências locais para rodar o software em outro ambiente e até envio de uma versão diferente para o ambiente de produção. Durante esse vídeo Ramon Durães discute o processo de Build e Release e a sua importância em um projeto DevOps.

O desenvolvimento de uma aplicação envolve algumas etapas até colocar o software em produção. O processo de Build e Release são fundamentais para construir e gerenciar a publicação em todos os ambientes. Em muitos projetos esses passos ainda são manuais que causam demora na publicação, falta de qualidade e de velocidade, pois muitas vezes dependem de várias pessoas e permissões para promover a versão nos ambientes colocando em risco inclusive o projeto e gerando a famosa frase “Na minha máquina funciona”.

Durante o planejamento de uma estratégia de modernização de aplicações um dos pontos críticos é a metodologia [DevOps](#) que aplicamos na gestão ágil dos projetos e cobrimos também o processo de Build / Release automatizando além da gestão do código incluindo a compilação e distribuição usando procedimentos padrões, automatizados e gerenciáveis para garantir uma operação de forma ágil, inteligente e segura.

Não são poucos os projetos que se promovem versões erradas inclusive existe um sério conflito de gestão de código e promoção de versão. Se você definiu um

processo de promoção de versão padrão como o modelo DEV -> QA -> Homologação > Pré-Produção -> Produção é fundamental garantir que a compilação "V1.0.1" passe por todos os ambientes seguindo todos os procedimentos de validação em cada ambiente para ao final caso aprovada seja a versão que vai ser realmente publicada em produção.

Nesse cenário apresentado a promoção é realizada do binário e você tem a garantia que todas as pessoas envolvidas na fila de homologação estão tratando da mesma versão "V1.0.1" e não de versões diferentes. Se você recompila em cada ambiente estará sujeito a problemas de merge, diferenças de versões e não poderá garantir com facilidade que a versão em produção é a mesma que passou em QA.

Se você achava que ser um excelente desenvolvedor era tudo então aproveite para rever os conceitos. O profissional de software precisa conhecer todo o ciclo da aplicação para ter certeza que o fluxo é único e integrado ponta a ponta senão você pode nem saber e/ou ter o mesmo código que foi usado para compilar a versão de produção. Para estranho isso para você? Já auditamos nos projetos de clientes diversos casos de alto risco como esse citado.

O Visual Studio da Microsoft traz um portfólio completo de [DevOps](#) tanto para uso interno na empresa quanto no modelo de SaaS em uma nuvem gerenciada pela Microsoft com o título de [VSTS](#) (Visual Studio Team Services) baseado no conceito já consolidado no mercado do [Team Foundation Server](#) (TFS). No contexto de [DevOps](#) você vai estabelecer a colaboração entre todos os pontos dos projetos e nesse cenário entra o controle de versões do código fonte ([TFS](#)/GIT) > Compilação (Team Build) > Publicação (Release Manager).

A nossa experiência em clientes no mercado com empresas de todos os tamanhos e verticais tem nos permitido observar o quanto de valor se gera a um projeto de software aplicando a metodologia [DevOps](#). É encantador receber o feedback do cliente sobre processos que ele tinha como críticos, demorados e não escaláveis e foram transformados passando o controle para os responsáveis pela promoção de depender de ninguém para garantir que a operação flua com velocidade e segurança. Se publicar uma versão é sempre um momento tenso pela possibilidade de bugs você deve imaginar essa tensão multiplicada por falhas de gestão de Build/Release.