

SCM och DevOps

Continuous Delivery har som begrepp mognat under de senaste åren och trenden är mycket tydlig:

– När organisationer blir mer och mer "agila" så skärps också kravet på en effektiv och löpande utveckling av program-, test- och målmiljö.

För att klara detta behövs tillförlitlighet, repeterbarhet och inte minst automation av utvecklingsprocessens alla delar. Att snabbt och säkert göra ändringar i test och produktion kräver utvecklings- och testmiljöer som, i så stor utsträckning som möjligt, efterliknar produktionsmiljön. Något som också minskar den risk som ofta förknippas med att låta nya utvecklare komma in i förädlingsprocessen. Grunden ska vara att ny kod integreras löpande för att snabbare och säkrare leverera ändringar till test och produktion.

Många organisationer har tagit **Continuous Delivery** ett steg vidare genom att tillämpa **Continuous Deployment**. Som resultat har begreppet DevOps dykt upp som en lysande stjärna inom SCM (Software Configuration Management).

Begreppet har inte fått en enhetlig tolkning än, men vanligtvis säger man att utvecklare (Dev) och drift (Ops) förstår och samarbetar med varandra för att effektivisera och säkra leveranserna. Vi kan se det som tekniskt upplägg där processerna görs repeterbara genom en hög grad av automatiserade metoder för byggen, installationer, tester och rapportering.

Värt att notera är att DevOps även används som beteckning för den roll som administrerar, underhåller och vidareutvecklar de tekniska systemen. Ett förhållande som liknar det vi kan se i rollen Configuration Manager och området Configuration Management.

Delivery Pipeline används ofta som gemensamt namn för en långt automatiserad miljö som ofta innehåller Automatiserade byggen, Continuous Integration, Automated Test, och Automated Deployment.

Med hjälp av virtuella miljöer kan antalet pipelines enkelt skalas upp genom parallellisering. Något som kan öka flödet i processen, throughput, och därmed korta ledtiderna, time-to-delivery.

Det här sättet att arbeta förutsätter att det finns en hög SCM medvetenhet hos alla inblandade. Systematiska fel kan snabbt skapa problem som är svåra att överblicka. Dessutom krävs djupare kunskap om hur de SCM-verktyg som används integreras på bästa sätt för att automatiseringarna skall fungera. Exempelvis måste det normalt finnas testade rutiner och färdiga skript för ovanligare konfigureringsaktiviteter om de behövs för att tillhandahålla den SCM funktionalitet man är beroende av.