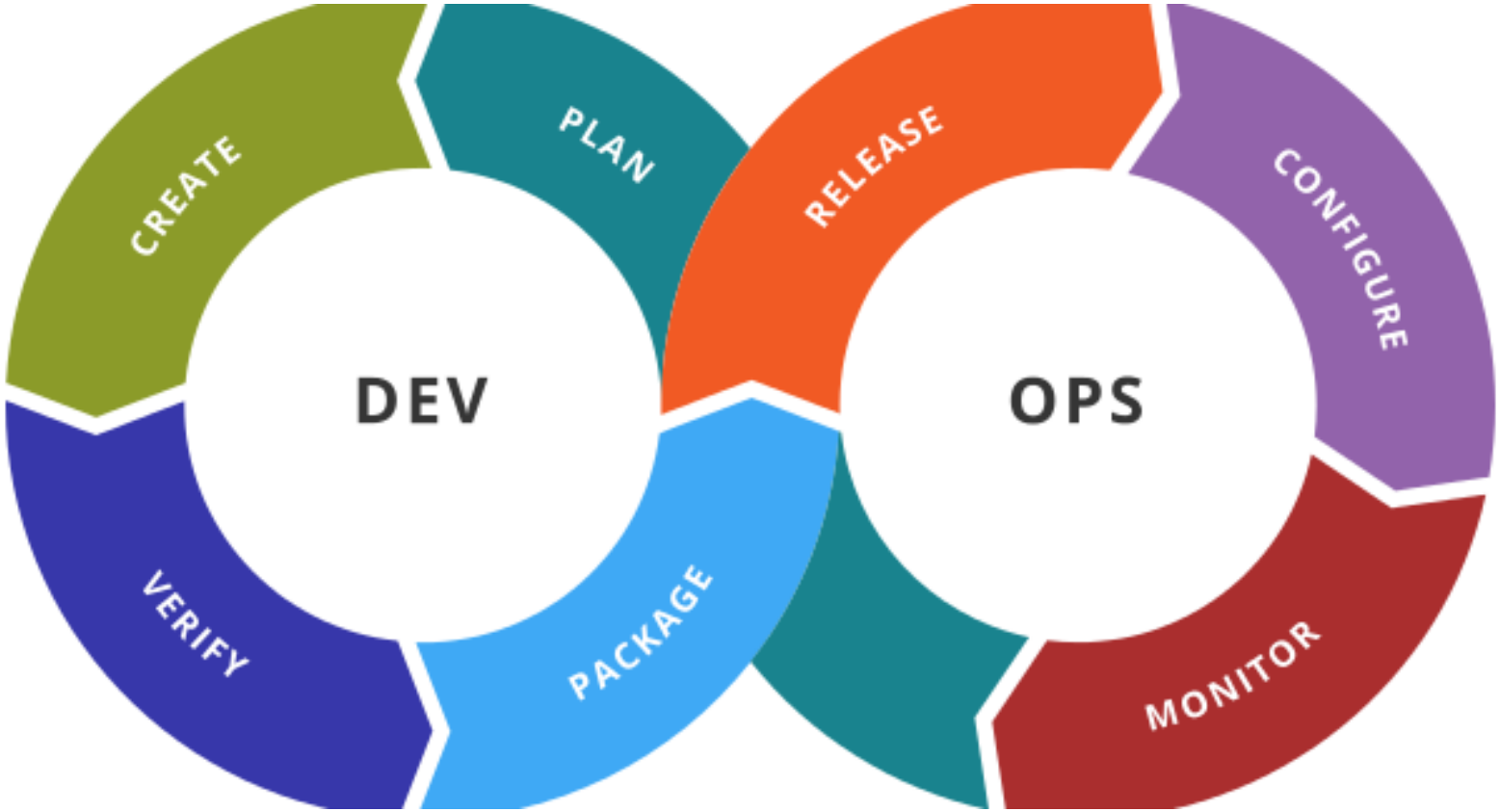
GIT DevOps

DevOps
Developers - Operations

DevOps innebär att samma team inte enbart utvecklar kod, utan även ser till att den deployas till produktion och körs.

Man brukar säga DevOps har sju livscykler där alla ska vara exakt lika delaktiga och kunniga på hela kodbasen.

Man arbetar då strikt efter det agila manifestet och det som man inte hinner under nuvarande arbetsfas(sprint), planeras in i nästa.


CREATE
PLAN
RELEASE
CONFIGURE
DEV
OPS
VERIFY
PACKAGE
MONITOR

Man arbetar mot samma branch i versionhanteringssystemet.

Antingen så låser man Master branch, använder sig av Pull Requests (PR), och mergar in dem efterhand de löser problem.

Eller så gör man extremt små commits, och pushar direkt till Master Branch, för att på så sätt registrera sina rader man arbetar i.

Om man arbetar på flera features, som kanske inte ska skeppas med i nästa release, skriver man en versionskontroll, så eventuella tester skippas.

Det är viktigt att alla checkar in sin kod i slutet av arbetsdagen. Detta gäller inte om man sitter med en för stor commit, då man oftast, inte alltid, skapar flertalet buggar för andra programmerare.

Jobbar man i ett team som sitter i flera tidszoner, pusha inte något halvfärdigt dagen innan semester / helg.

Är DevOps detsamma som CI / CD?

Strikt tolkning: NEJ

Inte strikt tolkning: NJA

Ibland låter man teamens CI del gå direkt till produktion, men detta kräver extremt välskriva smoke test, att hela produkten klarar av alla typer av interaktioner.

Oftast låter man en deployment gå till en s.k. deployment pipe, där vissa valda personer får testa applikationen.

När dessa test har pågått en viss tid, och inga oönskade fel har inrapporterats, sätts den i live produktion.

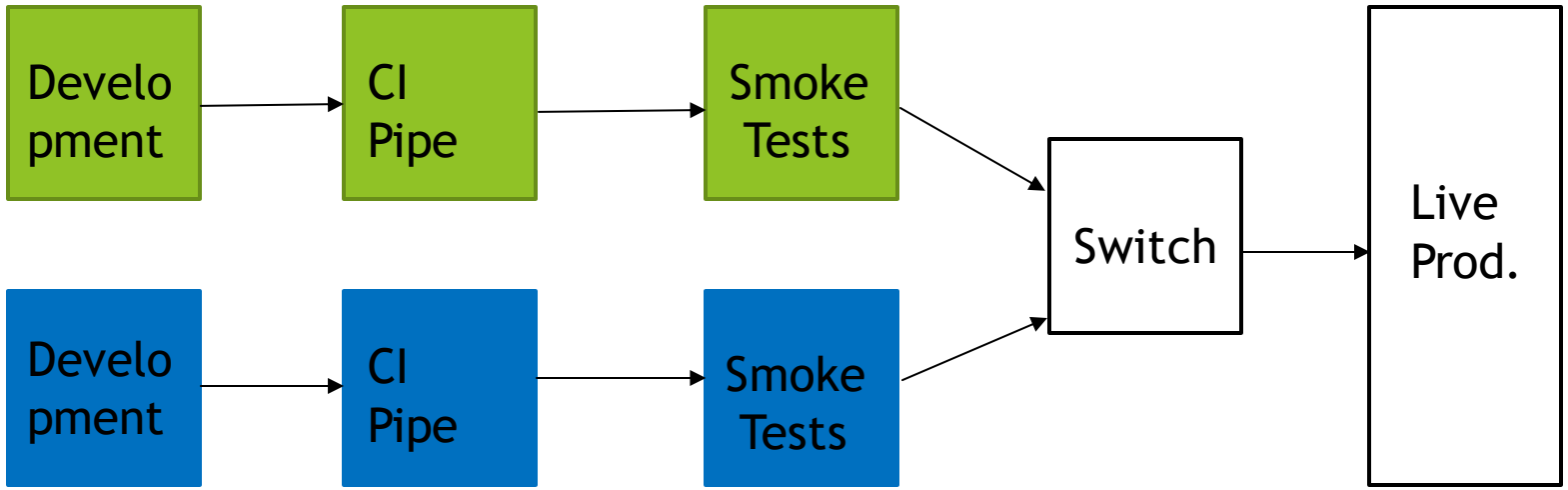
Nu kommer vi in på Blue – Green Deployment

Blue – Green Deployment

Den ena färgen är produktions deployment, medans den andra färgen är development.

Man utvecklar enbart emot development och när den är redo, gör amn en switch, så Blue blir development, och Green då Deployment.

Rätt konfigurerat minimerar detta downtime för applikationen.



På detta sätt så märker inte användaren något, och kan aldrig veta hur utvecklingen sker.

Ju mer automatiskt, desto mer tilltro till utvecklingslaget men desto större risk. Avvägningar....

Övningar

Utse en som skall vara Teamleader. Hen skapar ett Github repo, genererar fem text filer och bjuder in alla i klassen. Därefter tar alla och klonar ner repot, sätter upp så man kan pusha till repot, och sen skriver ni i filerna. Testa göra commits och pusches med intervaller utan att kommunicera med varandra.