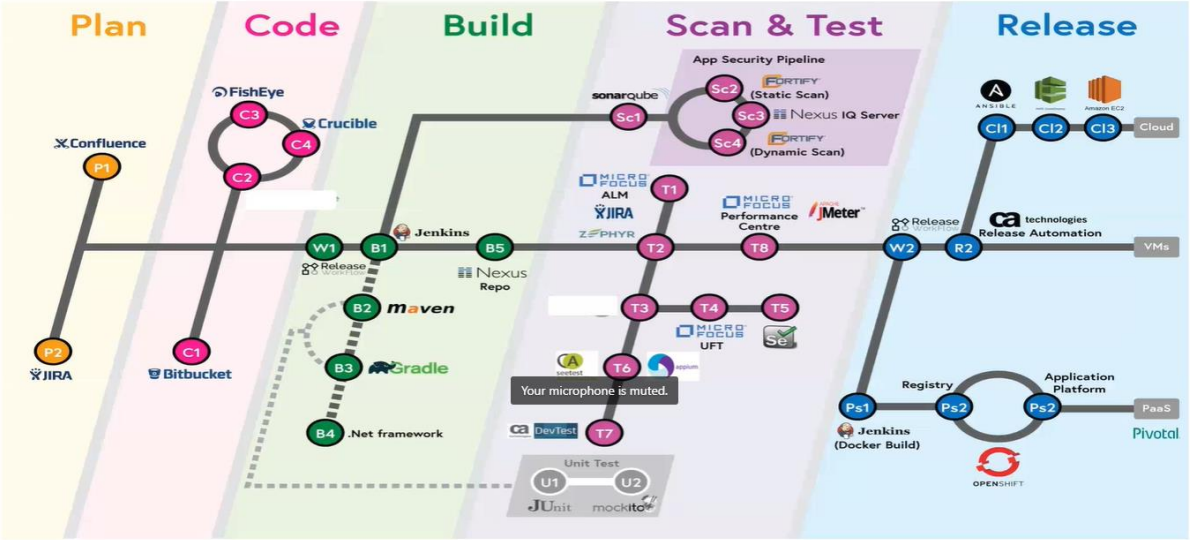
Enterprise DevOps Platform



SDLC2 Framework

What is SDLC2 ?

SDLC2 is a framework that consists the following .

- **Guidelines** e.g. how you branch and merge your code.
- **Principles** e.g. Fast-forward only merging, production ready main branch.
- **Pipelines** e.g. each type of application will have it's own application , promotion and release pipeline.
- **Guardrails** e.g. Automated checks in pipelines to ensure that the guidelines and principles are strictly adhered to.

Branching Models for CI/CD

- **Trunked Based Branching Model**
  1) master is the only long lived branch. Only fast-forward merging is allowed into master.
  2) tags with PROD<tag> represents what is running in production.
  3) master is where developers merge their production-ready code.
- **Integration Branching Model**
  1) master and integration are the only long lived git branches. Only fast-forward merging is allowed into these branches.
  2) master represents what is running in production now.
  3) Integration is the future version of master where developers merge their feature branches .

Fast-forward merging: branch has same no of commits as master

Image scanning : Aquasec,Twistlock

V5

SDLC2 Framework

General Benefits

- **Reduced Manual intervention** with simplified process
- **Decreased lead time** from source code push to production deployment
- Ensures "**production-ready**" and "**release-able**" main branch in bitbucket. Reduces long-lived branches.
- **Reduce** duplication or work e.g. pipeline creation. Increases re-usability
- **Scalability** – no more 1 pipeline per application , 100 pipelines for 100 applications
- **Default Blue-Green deployments** for non-DEV deployments.

For Developers

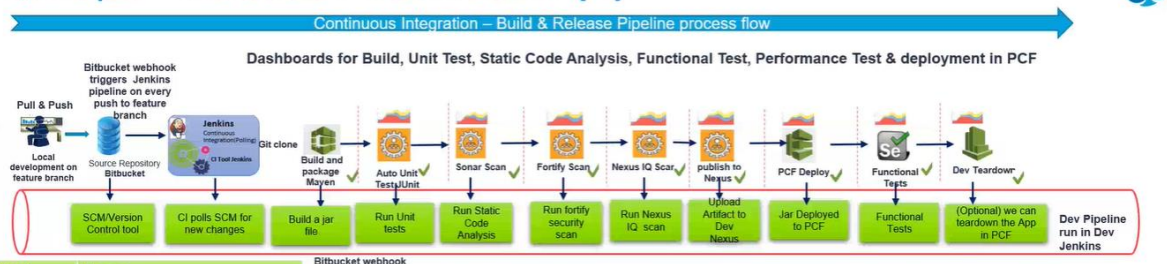
- **Frees up developer's time** from pipeline development to improve quality of application & performance .
- **Increased feedback** to developer e.g. pipeline will fail and report the exact issue it it doesn't meet the required standards .
- **Easy** for Application team to onboard .
- Ensures **code quality** standards (scans and validations done at multiple points of the process).
- Ensures **integrity** of artifacts (build only once in dev and promoted to production).

For Operators

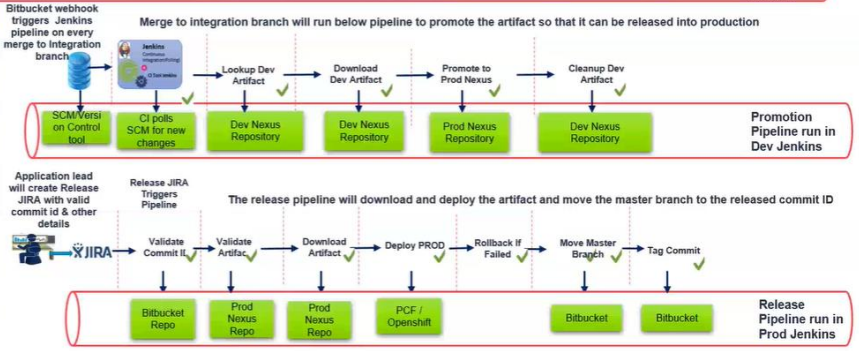
Your microphone is muted.

- **Ease** of maintenance and support. Standard set of pipelines Vs 1 application team 1 pipeline
- **Reduces** housekeeping tasks with dev and prod nexus and promotion concept.

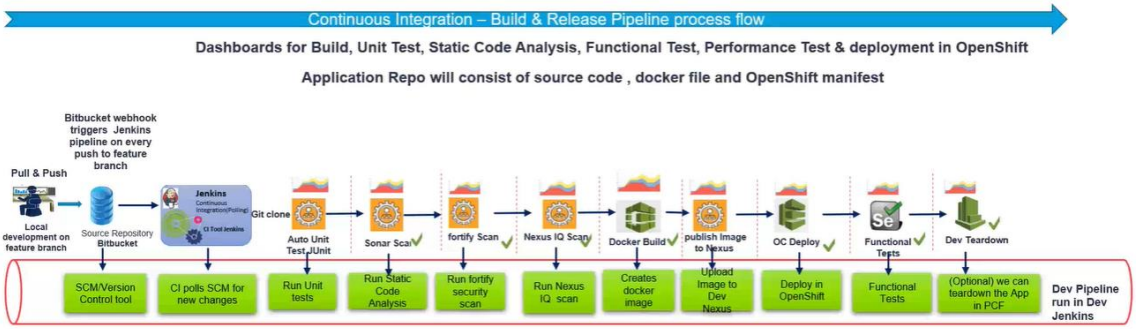
DevOps - Release Automation (1)



Tool	Usage
Jira	Change Management
Jenkins	CI/CD
JUNIT	Unit Test
Bitbucket	Source Code Management
SonarQube	Static Code Analysis
Fortify	Security Scan
Nexus IQ	Scan for Opensource Components
Nexus	Artifact Repository
PCF	Code Deployment (PaaS)
OpenShift	Container Orchestration (PaaS)
Docker	Containerization platform
Selenium	Functional Test



DevOps - Release Automation (2)



1:1 mapping of repo <> microservice
- 300+ repos currently managed in these pipelines
- Around 150 developers working on these

Your microphone is muted.

Deployment to UAT can easily be < 10 mins
- Did this more than 2000+ times in Sep 2019

DevOps Implementation using Agile Practices

- Scope and Product Vision is defined in Initiation Phase
- Epics and User Stories are developed by understanding Customer Personas & Journeys
- A story map (A Journey that the customer takes with the product) is prepared to develop the product backlog and release plan
- Prioritization and estimation of the user stories are done to determine the release plan / schedule
- Sprint Cycle (Duration is for 2 Weeks)
  - Backlog grooming / refinement session in the beginning of every sprint.
  - Elaborating on the user stories and refining User Story estimation.
  - Daily Scrum Call, where the current status, roadblocks and upcoming activity are discussed.
  - The burn down charts from JIRA are exported to know the amount of work left to do and the time available.
- At the end of the Sprint
  - Sprint Review and Demo to showcase the completed Product Backlog items selected for the sprint
  - Sprint Retrospective (Start Stop Continue Retrospective) at the end of sprint to consolidate learning.
  - The original estimate and actual efforts are logged in JIRA against the user story to improve on estimations
  - KPI's measured – Sprint Burndown Chart, Velocity, Automaton of Test Cases, Technical Debt Reduction etc.,

Roles	Responsible
Scrum Master	DBS
Product Owner	DBS
Developers	Capgemini
Testers	Capgemini
Business Analyst	DBS / Capgemini

DevOps + Agile – Business Benefits Delivered

- Build Management & Continuous Integration - Fully automated builds, dependency management, repository management and fully automated tests
- Deployment & Release Management - Fully automated deployments, automatic rollbacks based on test results. Matured process in Dev/SIT env to manage build and deployment related issues
- Environment Maintenance & Incident Handlings - Stability has improved significantly. Developers knows their common mistakes for env breaking
- Proactive Monitoring, alerts and health checks - Fully automated monitoring enables alerting mechanisms which triggers emails, other proactive communications round the clock. Automated preventive measures for repeated or known problems
- Creating single and unified teams for the full development lifecycle. Applying Agile principles across the enterprise
- Automated deployment of applications across development, test, staging and production stages
- Cultural Benefits – Happier and more productive teams, Higher employee engagement, more time to innovate rather than fix and maintain

DevOps assessment on heals solution:

Capgemini Recommended View – CI/CD Pipeline

[Tools to be evaluated based on the existing applications on AWS Cloud]