Software Configuration Management in a DevOps context

Lars Bendix
bendix@snescm.org

Scandinavian Network of Excellence in Software Configuration Management (sneSCM.org)

Christian Pendleton
cpe@praqma.net

Praqma AB
Malmö
Sweden

http://cs.lth.se/~bendix/Research/SCMnDevOps/

Preamble

Premise:

- no project can be (successfully) carried out without SCM
- it might not be called SCM
- it might not be carried out by SCM people
- so, from the outside it might seem like SCM is absent in DevOps

Research Questions

RQs: What are the relations between SCM and DevOps?

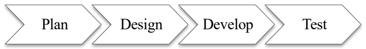
- what things from traditional SCM are not needed in a DevOps context?
- what new thing need to be added to the SCM toolbox?
- how should "old principles" be cast in a DevOps context?
- how could SCM sell itself to DevOps?
- why should DevOps buy SCM?

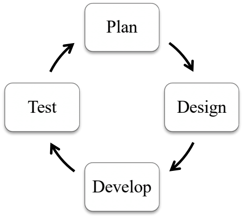
Agenda

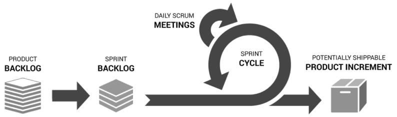
Agenda:

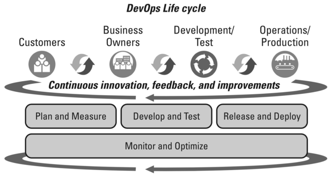
- Motivation
- What is DevOps?
- State of DevOps analysis
- SCM proposals
- Findings

SCM needs to adapt









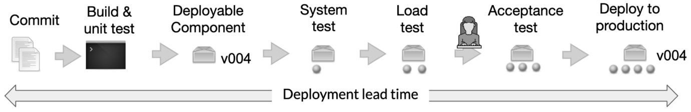
“Research Method”

Investigating:

- literature study
- CMCMs (Malmö & Rome)
- brainstorming sessions
- DevOpsDaysCPH Open Space
- DevOpsMalmoe Meetup
- ”presenting”:
  - Italian SCM summit
  - BCS CMSG conference
  - Scandinavian SCM day

What is DevOps – short version

DevOps is: “a set of practices intended to **reduce the time** between **committing a change** to a system and the change being placed into normal **production**, while ensuring **high quality**”



Also includes **monitoring** and **measuring** of the software in runtime!

What is DevOps – longer version I

Culture (People, processes and tools)

Automation (Do it once, do it twice, automate)

Lean (Continuous improvement and learning)

Measurement (development, production, business)

Sharing (Collaborate, give feedback, don’t copy)

What is DevOps – longer version II

Goals:

- improved deployment frequency
- lower failure rate of new releases
- shortened lead time between fixes
- faster mean time to recovery

Aims to maximize the **predictability**, **efficiency**, **security** and **maintainability** of operational processes.

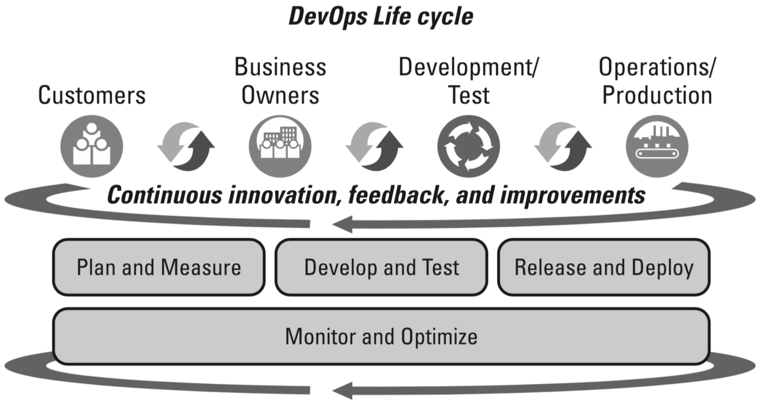
What is DevOps – longer version III

How does DevOps differ from Agile/Scrum/XP?

What more is DevOps than “just”: Dev + Ops?

Idea => backlog => development => production => used by people

In order to get as fast as possible from idea to use we take a **small** (part of an) idea and work **continuously** on it in a **cross-functional** way until it is in production and then we **monitor** what happens when it is used to get **feedback** that creates new small ideas that ……



DevOps Life cycle

Customers

Business Owners

Development/ Test

Operations/ Production

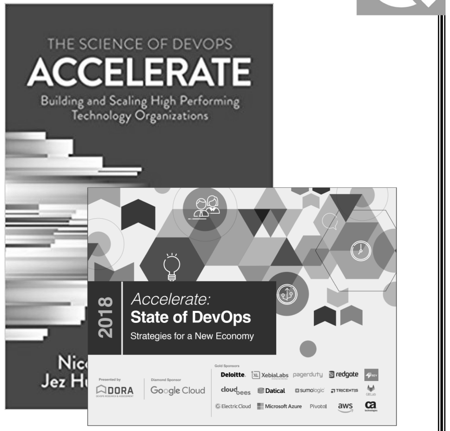
Continuous innovation, feedback, and improvements

Plan and Measure

Develop and Test

Release and Deploy

Monitor and Optimize


THE SCIENCE OF DEVOPS
ACCELERATE
Building and Scaling High Performing Technology Organizations
Nic
Jez H
2018
Accelerate:
State of DevOps
Strategies for a New Economy
DORA
Google Cloud

DevOps evolution in stages

The five stages of DevOps evolution:

- Stage 0: Build the foundation (facilitate sharing of ideas, metrics, knowledge, processes and technologies)
- Stage 1: Normalize the technology stack (agile, version control, continuous integration)
- Stage 2: Standardize and reduce variability (reduce overall complexity, reduce errors from inconsistencies)
- Stage 3: Expand DevOps practices (deployments are a huge pain, provide predictability and reliability)
- Stage 4: Automate infrastructure delivery (automation of system configuration and provisioning)
- Stage 5: Provide self-service capabilities

SCM-related DevOps activities I

Foundational practices:

- Monitoring and alerting are configurable by the team operating the service
- Deployment patterns for building applications or services are reused
- Testing patterns for building applications or services are reused
- Teams contribute improvements to tooling provided by other teams
- Configurations are managed by a configuration management tool

SCM-related DevOps activities II

Various practices:

- Application development teams use version control
- Put application configurations in version control
- Put system configurations in version control
- Infrastructure teams use version control
- Source code is available to other teams
- Teams use continuous integration
- System configuration is automated
- Provisioning is automated
- Success metrics for projects are visible

SCM-related DevOps activities III

Automate system configurations

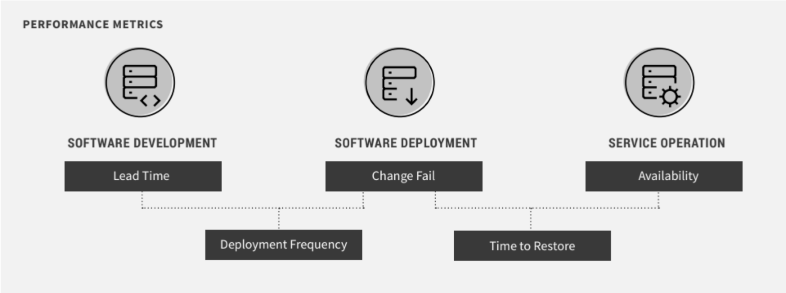
- Some teams have a goal of automating all change, giving them completely repeatable, rebuildable systems.
- Consistency: Automated tasks follow a set process and thus produce predictable results.
- Documented behavior: Tasks now have a defined way they are supposed to work, so are easier to troubleshoot.

Application configurations are in version control

- Should be versioned, auditable, contain history, and ideally, the reasons why they changed.
- Separating code from configuration data allows for more rapid deployments, updates and validation.

Five metrics define SDO performance

Software delivery and operational performance:



PERFORMANCE METRICS

SOFTWARE DEVELOPMENT

Lead Time

SOFTWARE DEPLOYMENT

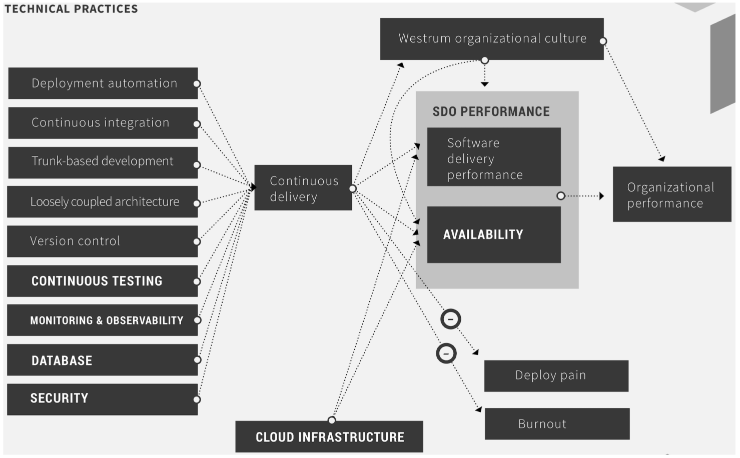
Change Fail

SERVICE OPERATION

Availability

Deployment Frequency

Time to Restore



TECHNICAL PRACTICES

Deployment automation

Continuous integration

Trunk-based development

Loosely coupled architecture

Version control

CONTINUOUS TESTING

MONITORING & OBSERVABILITY

DATABASE

SECURITY

CLOUD INFRASTRUCTURE

Continuous delivery

Westrum organizational culture

SDO PERFORMANCE

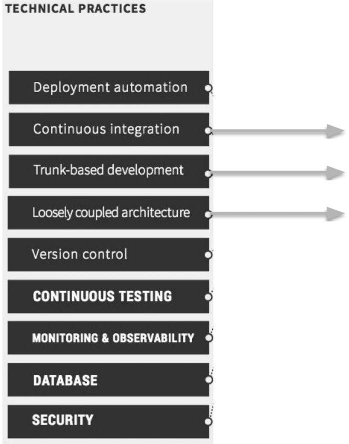
Software delivery performance

AVAILABILITY

Organizational performance

Deploy pain

Burnout



TECHNICAL PRACTICES

Deployment automation

Continuous integration

Trunk-based development

Loosely coupled architecture

Version control

CONTINUOUS TESTING

MONITORING & OBSERVABILITY

DATABASE

SECURITY

Frequent commits

No long-lived feature branches

"… it's easy to modify or replace any individual component or service"

Test & infrastructure managed

Test data is available

Get feedback from automated tests in less than ten minutes

Continuously *review and improve* test suites to better find defects and keep complexity and cost under control

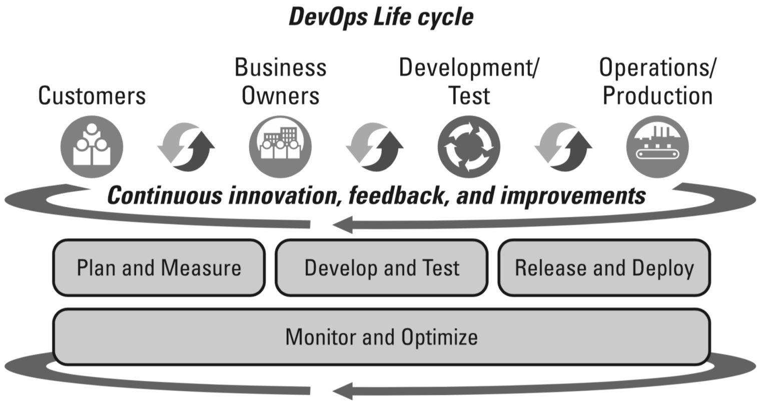
What is DevOps – longer version III

How does DevOps differ from Agile/Scrum/XP?

What more is DevOps than “just”: Dev + Ops?

Idea => backlog => development => production => used by people

In order to get as fast as possible from idea to use we take a **small** (part of an) idea and work **continuously** on it in a **cross-functional** way until it is in production and then we **monitor** what happens when it is used to get **feedback** that creates new ideas that ……



SCM for Agile I

SCM-related Agile practices:

- Collective Code Ownership
- Continuous Integration
- Test-Driven Development
- Frequent Releases
- Planning Game
- Refactoring

SCM for Agile II

SCM sub-practices for Agile:

- use a version control tool
- define configuration items and their structure
- use a build tool
- automate and optimise the Release process
- physical audit in Release process
- impact analyse stories during Planning Game
- impact analyse refactorings
- incremental refactoring
- use a copy-merge work model
- keep the repository clean
- write proper commit comments
- trace changes to stories

SCM for Dev + Ops (=CoDe?)

SCM sub-practices for CoDe:

- automate deploys
- create pipelines
- provision environments
- build quality gates
- separate code and configuration data (one build)
- architecture (microservices):
  - high cohesion
  - low coupling
  - traceability
  - selected testing

SCM for DevOps

SCM sub-practices for DevOps:

- Plan & Measure:
  - CCB
  - CSA?
- Monitor & Optimize:
  - A/B testing
  - canary releases
  - variants
  - CSA?

SCM technical debt in DevOps

Things that DevOps people do know about:

- git
- Jenkins
- Jira
- …

Things that DevOps people do **not** know about:

- CI
- CMDB
- baselines
- BoM
- CR & CCB
- …

Findings I

DevOps change requests to SCM:

- help us – please ;-)
- handle many small changes – roughly one day to one week/task
- keep us in the flow – once we stop we are dead
- make it fast – we hate to wait
- KISS – because **we** will do it (as instructed)

Findings II

SCM as a discipline is invariant (is published)
- concepts and principles

SCM as a role will change (will not perish)
- operational => automated
- operational => ”out-sourced”
- strategic => teaching (deputy)?

“Research Method”

Searching further:
- brainstorming sessions
- literature studies
- DevOps CPH Meetup
- interviews:
  - ”zero to DevOps” companies
  - ”hardware to DevOps” companies
- questionnaire
- white paper

References

Software Configuration Management and Continuous Software Engineering: http://fileadmin.cs.lth.se/cs/Personal/Lars_Bendix/Research/SCMnCSE/

Lars Bendix, Torbjörn Ekman: *Software Configuration Management in Agile Development*, in the book Agile Software Development Quality Assurance, February 2007.

Ulf Asklund, Lars Bendix, Torbjörn Ekman: *Software Configuration Management Practices for eXtreme Programming Teams*, in proceedings of the 11th Nordic Workshop on Programming and Software Development Tools and Techniques - NWPER'2004, Turku, Finland, August 17-19, 2004.

Ulf Asklund, Lars Bendix, Torbjörn Ekman : *Configuration Management for eXtreme Programming*, in proceedings of the Third Conference on Software Engineering Research and Practise in Sweden, Lund, Sweden, October 23-24, 2003.