



What SCM is useful and necessary on DevOps projects?

Lars Bendix

bendix@snescm.org

Christian Pendleton

cpe@pragma.net

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

*Praqma AB
Malmö
Sweden*

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

© Lars Bendix & Christian Pendleton – sneSCM.org

Oslo University, Norway, October 2, 2019



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

© Lars Bendix & Christian Pendleton – sneSCM.org

Oslo University, Norway, October 2, 2019



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?

© Lars Bendix & Christian Pendleton – sneSCM.org

Oslo University, Norway, October 2, 2019



Agenda

Agenda:

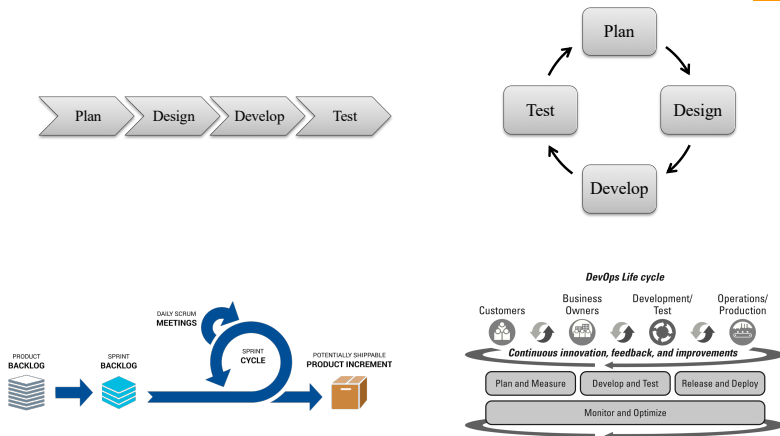
- Motivation
- What is DevOps?
- What is SCM?
- SCM for DevOps findings

© Lars Bendix & Christian Pendleton – sneSCM.org

Oslo University, Norway, October 2, 2019



SCM needs to adapt



“Research Method”



Investigating:

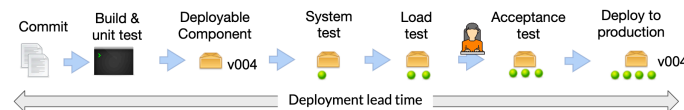
- literature study
- CMCs (Malmö, Copenhagen & Rome)
- brainstorming sessions
- DevOpsDaysCPH Open Space
- DevOpsMalmoe Meetup
- ”presenting”:
 - Italian SCM summit
 - BCS CMSG conference
 - Scandinavian SCM day



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?

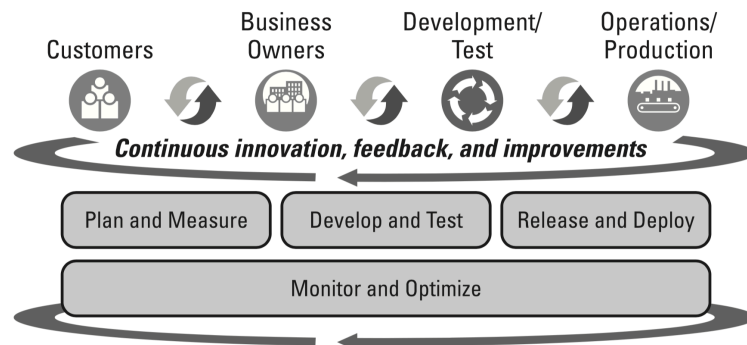
What more is DevOps than “just”: Dev + Ops?

Is it “monitoring and measuring users” to know what they like?

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



What is SCM – short version



Software Configuration Management is this cool stuff that will facilitate a team to coordinate people and things so they can carry out changes in an orderly fashion right from idea/conception to production – and retirement – and avoid any chaos and confusion in the process.

SCM will make sure that you know exactly what you have, not just when it is in production but also while you are developing it and SCM will provide a project with all the quality gates they could desire for.

SCM can be done in formal and rigid ways (top-down/waterfall) - or it can be done in more informal and flexible ways (bottom-up/Agile, DevOps).



What is SCM – longer version I



It is about tracking, managing and controlling **changes** to (re-)establish **baselines**.

A process for establishing and maintaining **consistency** and **integrity** of a product. Provides **visibility**, control, orderly change.

Provisions for the **storing**, tracking and updating of **all parts**.

Tracks requirements (change requests) **throughout the life-cycle**.

Establishes baselines and performs a standard change management process through to “release management and delivery”.

Configuration management database – Configuration Items, their attributes and the dependencies between them.



What is SCM – longer version II



Part CM:

- Identification (configuration items, baselines)
- Control (change process, change requests, CCB)
- Status Accounting (recording and reporting information)
- Audit (assesses compliance with requirements before acceptance of change into a baseline)

Part Software:

- Build management
- Process management (adherence to development processes)
- Environment management
- Teamwork (facilitate team interactions)
- Defect tracking



What is SCM – longer version III



- Coordination problems: SD, DM, SU
- Coordination strategies: locking, copy/merge, LT/SLT
- branching strategies
- history and diffing
- CI, CMDB, traceability, baselines, BoM
- CSA
- CR, change process, CBB
- CA



SCM for DevOps findings – I



Different types of SCM:

- Strategic SCM
- Operational SCM

Different roads to DevSCMOps:

“Zero” → DevOps&SCM

”Hardware” → DevOps&SCM

“Zero” → DevSCMOps ← “Hardware”



SCM for DevOps findings – II



How to organize SCM:

- Company (usually start-up) with only one team (3-8 people):
 - SCM as a service / consultant + ”deputy”
- Company with 3-4+ teams:
 - SCM “team”
 - SCM on teams



SCM for DevOps findings – III



DevOps – mostly – has three parts/ingredients:

- agile in some way
- Dev + Ops – together or cross-functional
- “monitor and measure”

and we need SCM to service and support each part/context



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 (dependencies)
 - selective 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
- ...



Conclusions 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)



Conclusions 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)?