What SCM is useful and necessary on DevOps projects?

Lars Bendix
bendix@sneSCM.org

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

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

Acknowledgements

Christian Pendleton
christian.pendleton@eficode.com

Eficode/Praqma AB
Malmö
Sweden

Erik Hochbergs
erik.hochbergs.175@student.lu.se

Laroy Nilsson Sjödahl
fpr08lni@student.lu.se

Department of Computer Science
Lund University
Sweden

Preamble

Premise:

- no project can be (successfully) carried out without (S)CM
- it might not be called SCM
- it might not be carried out by SCM people
- but – from the outside it seems like SCM is absent in DevOps

Preamble

Premise:

- no project can be (successfully) carried out without (S)CM
- it might not be called SCM
- it might not be carried out by SCM people
- but – from the outside it seems like SCM is absent in DevOps – it is **NOT**!

Research Questions

RQs: How to carry out SCM in a DevOps context?

- what things from traditional SCM are *not needed* in a DevOps context?
- what *new things* need to be added to the SCM toolbox?
- how should ”old principles” *be cast* in a DevOps context?

Presentation objectives

After this presentation you will:

- know what role SCM has (also) in DevOps
- have a better idea of where you need SCM activities in your DevOps processes
- understand what SCM activities you actually carry out currently

Agenda

Agenda:

- What is DevOps?
- What is SCM?
- SCM for DevOps findings
- Q&A

What is DevOps – short version

I don’t know ;-)

Why DevOps?

Spotify:

- they don’t know their customers
- they can’t talk to their customers
- they can ***monitor*** what their customers ***do***
- they want to bring ***ideas*** into production fast
- **if** something catches the customers’ attention:
  - it is further developed/explored
- **if not**:
  - it dies
- it is a ”requirements engineering” method
- with emphasis on ”***fast from idea to production***”
- **fast from question to answer**

What is DevOps – longer version

Automation

Sharing

Continuous Integration

Feedback loops

Shift left

Continuous Testing

Continuous Delivery

Infrastructure as Code

Continuous Deployment

Culture

Monitor

Change management

Collaboration

Quality Assurance

Planning

Agile & Lean

from [Hochbergs, Sjödahl, 2020]

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.

[https://en.wikipedia.org/wiki/DevOps]

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

Configuration Identification
Configuration Status Accounting
Configuration Control
Configuration Audit
SCM plan

Build management
Release management
Change management
Version control
Teamwork
Workspace management

from [Hochbergs, Sjödahl, 2020]

SCM for Agile I

SCM-related Agile practices:

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

from [Bendix, Ekman, 2007]

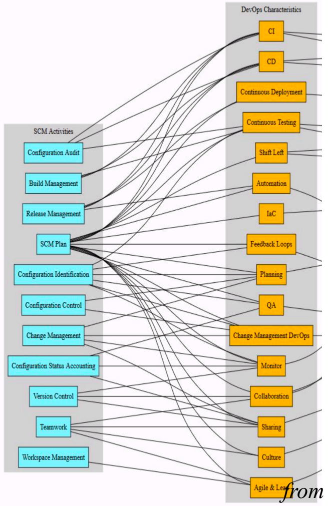
SCM for Agile II

SCM sub-practices for Agile:

- use a version control tool
- use a build tool
- define configuration items and their structure
- 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

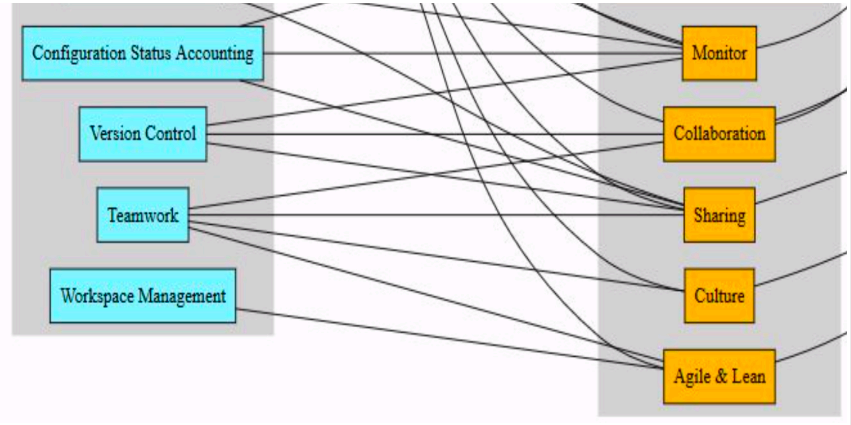
from [Asklund, Bendix, Ekman, 2004]

SCM in DevOps I



from [Hochbergs, Sjödahl, 2020]

SCM in DevOps II



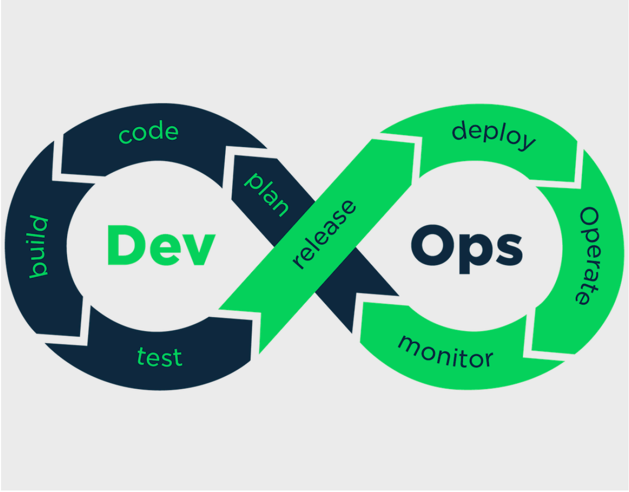
from [Hochbergs, Sjödahl, 2020]

SCM for DevOps (=CD?)

SCM sub-practices for CD:

- 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

Initial analysis



Initial analysis

Monitor:

- define metrics to monitor
- collect and process data
- => proposed Change Requests

Plan:

- Change Requests from ”all sources”
- ”Change Control Board”
- => Change Requests to be implemented

Initial analysis

Code:

- handling parallel work
- Continuous Integration
- => source code to build

Build:

- automation of build process
- identification of build (BoM)
- => systems to be tested

Initial analysis

Test:

- test automation
- test selection
- => verified system

Release:

- packaging of release
- configuration audit
- => system to be deployed

Initial analysis

Deploy:

- system configuration
- deployment automation
- => deployed system

Operate:

- ???
- ???
- => data?

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

[Puppet, 2018]

Summarizing

Research answers:

- all SCM concepts and principles are useful in a DevOps context
- except for "SCM Plan" – and then…
- implement SCM ”light-weight" and on "as need basis":
  - agile aspects
  - "continuous" integration/delivery/deployment/testing/…
  - automation? do it once, do it twice – automate!
  - quality gates – "configuration audits"

Conclusions

- SCM is *not* "one size fits all"
- SCM does *not* have the be "rocket science"
- indications are that SCM is an important supporting activity also for DevOps projects

References

[Hochbergs, Sjödahl, 2020]: Erik Hochbergs, Laroy Nilsson Sjödahl: *Software Configuration Management in a DevOps context*, Master's thesis, Lund University, Sweden, January 2020.

[Asklund, Bendix, Ekman, 2004]: 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.

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

[https://en.wikipedia.org/wiki/DevOps]

[Puppet, 2018]: Andi Mann, Michael Stahnke, Alanna Brown, Nigel Kersten: *State of DevOps Report 2018*, Puppet white paper, https://puppet.com/resources/report/2018-state-devops-report/