Taming Software Development Complexity via Reversibility

Claudio Antares Mezzina
Università di Urbino (Italy)

ICE@DISCOTEC2020 (virtually in Malta)

Reversibility (thermodynamics)?

A reversible process is a process whose direction can be returned to its original position by inducing infinitesimal changes to some property of the system via its surroundings

The level of entropy of a system is minimised when a process is nearly reversible

There is a tight connection between reversibility and entropy

Entropy

- Entropy measures the degree to which the probability of the system is spread out over different possible micro-states
- Entropy is related to the number of internal (or micro)-states that a system can have
- If we take this level of definition, and apply it to information theory, then we could say that entropy is related to the number of bits that are necessary to describe the state of a system
- Therefore, the more complex a system is, the more bits we need to describe it

Entropy, and hence system complexity, is minimised when a system is reversible

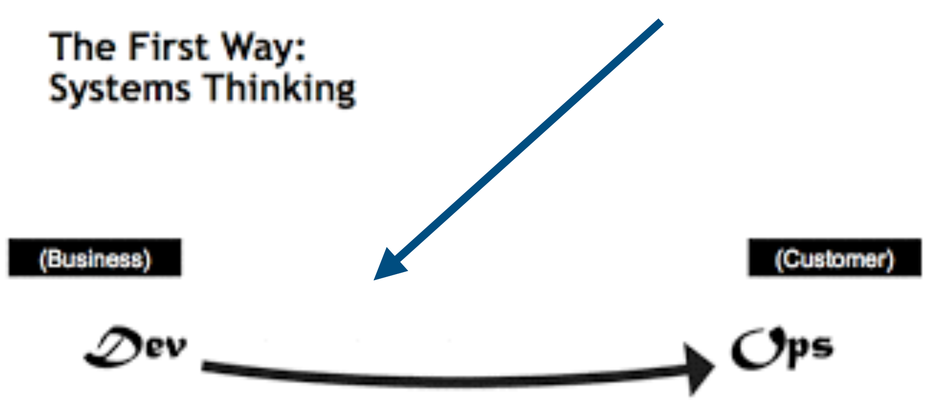
Stephen Lower: *Thermodynamics of Chemical Equilibrium*

Software Development DevOps

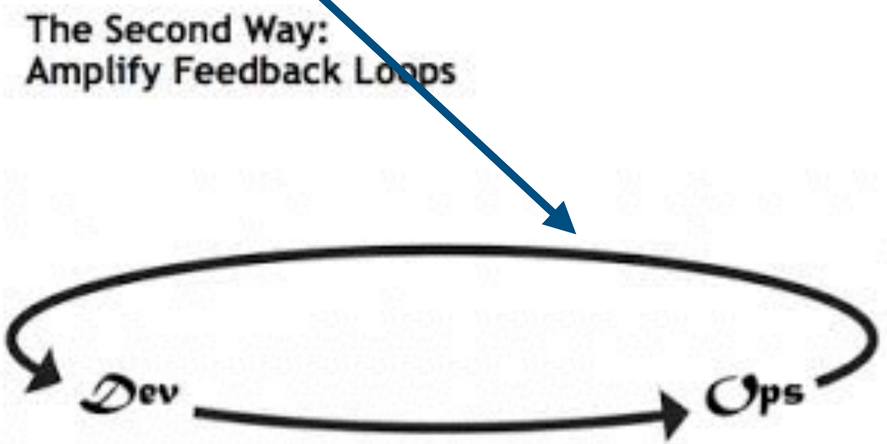
- Very tedious and complex process
- Coordination and management of different activities
- DevOps is a novel approach to software delivery (Developers + Operations)
- Involves two techniques: Continuous Integration and Continuous Delivery

DevOps

Irreversible and complex process

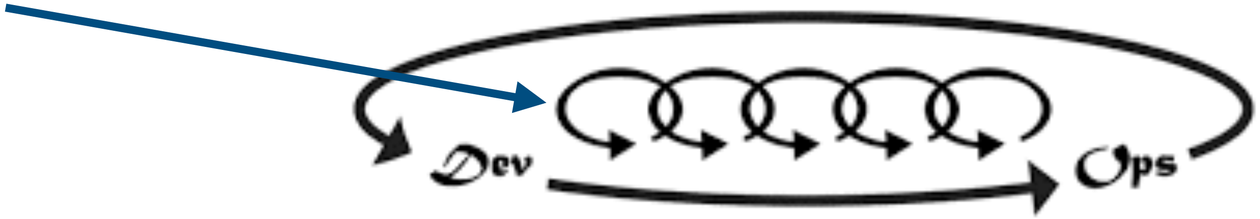


Reversible but complex process



The Third Way:
Culture Of Continual Experimentation And Learning

Reversible and simple process



Reversible DevOps?

- One of the CAMS principle of DevOps is Automation
- The deployment workflow should be automated and treated as part of the software being developed
- CI/CD pipelines allow to automate such phase
- This calls for pipelines and scripting language which support natively reversibility

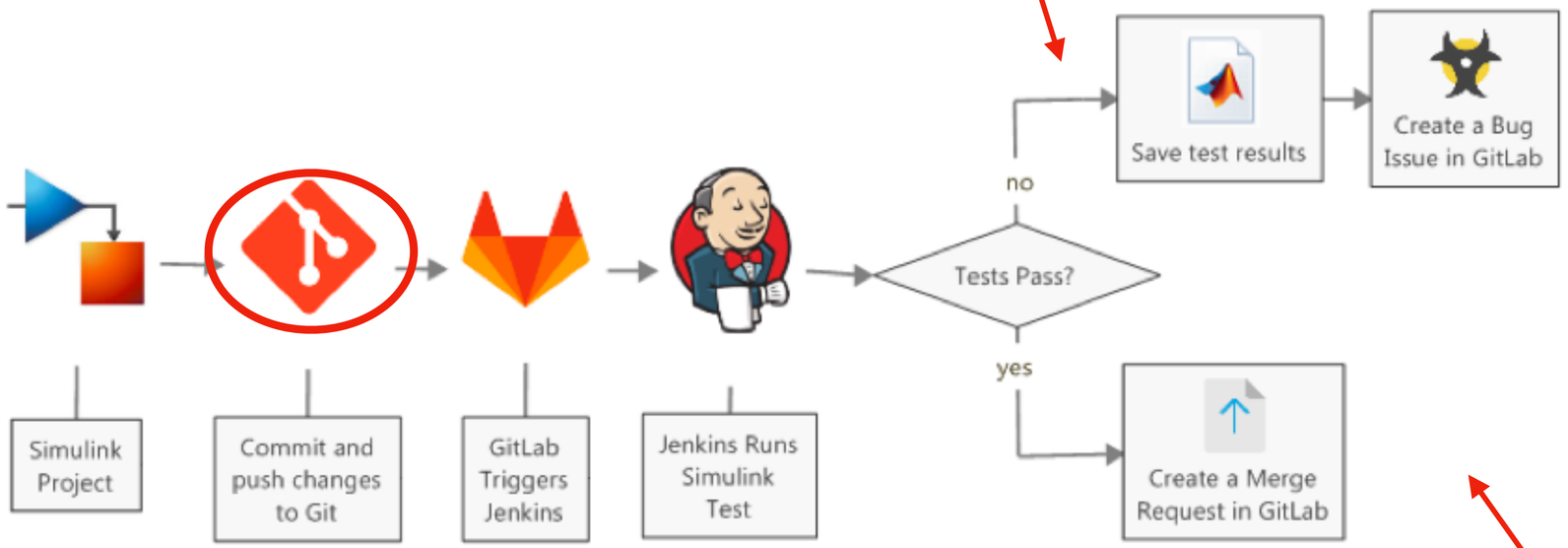
Where to reverse?

- Adding an automatic staging policy for commits could be an idea (in order to rollback to a previous state)
- If deployment fails, then automatically the previous artefact should be put back (sort of rollback/recovery)
- Quite often a failed test leave the system in an unstable state which has to be fixed in a manual way

Where to reverse?

Failed tests leave the system in an unstable state

Tagging strategy in order to rollback to previous states



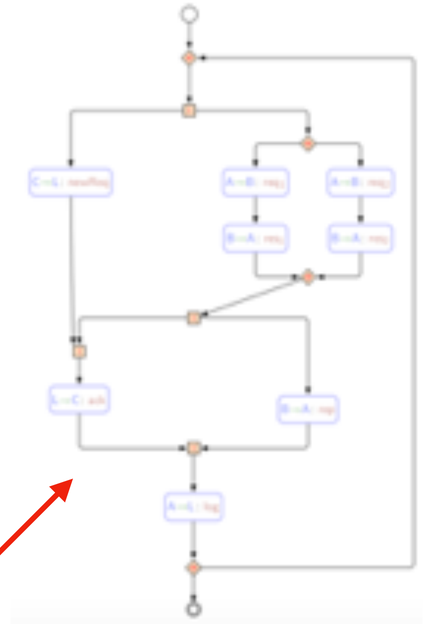
If deployment fails we have to restore the previous artefact

Reversible Global Graphs

Reversible Choreographies via Monitoring in Erlang

Adrian Francalanza[1], Claudio Antares Mezzina[2], and Emilio Tuosto[2]

[1] University of Malta, Malta adrian.francalanza@um.edu.mt
[2] IMT Advanced Studies Lucca, Italy claudio.mezzina@imtlucca.it
[3] University of Leicester, UK emilio@le.ac.uk



Graphical way to describe a reversible pipeline

Automatic scripting generation

Conclusions

- Reversibility can be seen as a tool to decrease system complexity
- DevOps reduces soft development to a sequence of small repeatable loops
- Small loops can be reverted in an automatic way by introducing reversibility into pipelines
- This calls for a new generation of automated tools for soft dev which natively supports reversibility
  - Starting from reversible global graphs