DevOps

A way to reduce risks for IoT?

Hui Song, SINTEF
Internet of Things

Ryan Manship, The Business of Federal Technology
As the IoT grows, so do the risks

- "I try to avoid all the risks before I go" – hardware thinking
  - Certification
  - Good design
  - Thorough testing

- "Hmm, we will see..." – software thinking
  - Keep changing
  - Continuous risk management
  - Prompt reaction
DevOps: The state of the art of software development practice

Up to 50 releases per day...
How software developers deal with risks

A short (simplified) history of software development models
It all started from "waterfall"
Agile development

Automatic testing

Software products
DevOps

Software Services
Operation is different in cloud
A developer's view of the history

I can code

I can also test it...

I can even operate it

Automatic test cases

Automatic deployment
Automatic monitoring

Do it often, automatically, and reproducibly

Waterfall  Agile  DevOps

OKAY. BUT HOW?
How comes DevOps

• Driving force: Ever-changing requirements and environments
• Actually required: Software as a service
  • More control of the software lifecycle
  • Even more fine-grained modularity (microservices)
  • Small teams
• Enabled by: New technologies
  • Cloud and containers -> Reproducible deployment
  • Framework and high-level languages -> actually readable code
  • Continuous integration pipelines
DevOps

DevOps is a set of practices (2) intended to reduce the time (1) between committing a change to a system and the change being placed into normal production, while ensuring high quality (3).

(1) Reduced time: relatively: 0, absolutely: up to 50 times a day*
(2) Automatic: everything is code, therefore programmable and tracked
(3) Without sacrifice of quality: thorough testing, chaos engineering...

How does it look like? A use case from Etsy

- Making many small, continuous changes: "how comfortable am I with deploying a change right now?"
- Every developer a Virtual Machine, configured by Chef, with the same cookbooks used in production
- Try is a tool that allows a developer to test his changes in Jenkins, without having to commit to trunk.
- The CI cluster is powerful enough to support 150 engineers, and more than 14000 tests suites runs per day.
- The pipeline passes through the staging environment, the same production environment, but only Etsy’s employees have access to it. One-click deployments by Deployinator
- Config flags supports completely enable or disable a feature or variants of a given feature, and thus allows A/B testing
- Developers do their own feature monitoring and everyone has access to all the graphs through dashboards.
- IRC is the main communication tool
Tools are the backbones to DevOps
DevOps and risks

- Use tools to minimize the known risks
  - Automation, traceability, reproducibility, testing...
- So that developers can handle the unknown risks quickly and continuously
More about DevOps tools

• No tools, no DevOps
• Tools work with each other in a flexible way
• Every team needs their own tools
DevOps in IoT

Fill-in the gaps
DevOps for IoT

- DevOps is far from being adopted in the IoT world
- Opportunities:
  - Fast to market, new devices and requirements, experimental culture...
  - A pragmatic way towards trustworthiness
- Challenges:
  - Hardware culture: certification, technical complexity, etc.
  - More unpredicted running environments
- Tool or Practice: a "chicken or the egg" problem:
  - A research and innovation action to build the missing tools
  - Demonstrate the practice on industrial use cases
The ENACT project

• Start date: 01-01-2018
• Duration: 3 years
• Project leader: SINTEF
• Consortium: 11 partners
DevOps for Trustworthy Smart IoT Systems

- Four high-level user stories: As a developer, I want to...
  - develop new and trustworthy features into an SIS in an agile way, so that I can promptly and continuously bring new values to the end users
  - release the changes into the running smart IoT system automatically and reproducibly, so that I can immediately see the effect of the changes
  - patch the running IoT system promptly and continuously, so that it keeps trustworthy despite internal threats (such as insufficient security setting, unexpected attacks, system failures, unexpected actuation conflicts, etc.)
  - patch the running IoT system promptly and continuously, so that it keeps trustworthy despite external threats (new vulnerabilities, software updates, new security strategies, new user profiles, policies changes, etc.)
Tools

- ThingML
- GeneSIS
- Actuation conflict management
- Diversifier
- GeneSIS
- Actuation conflict management
- Test, simulation And emulation
- Test, simulation And emulation
- Risk Management
- GeneSIS
- Diversifier
- Context-aware Access control
- Security and Privacy control
- Behavioral drift analysis
- Security and privacy monitoring
- RCA
A sample story

Rail Domain

Smart Building

eHealth

Context-aware Access control (Configure context)

Security and privacy monitoring

RCA

Diversifier (choose one architecture)

GeneSIS

Diversifier (alternative architectures)

Risk Management

GeneSIS

Test, simulation And emulation

Test, simulation And emulation

CODE

PLAN

RELEASE & DEPLOY

OPERATE

MONITOR

DEV

OPS
Conclusions

- DevOps facilitates agility
- DevOps makes real-time risk management possible
- IoT requires DevOps
- Currently some key tools are missing for IoT DevOps
- At SINTEF, we are trying to deal with this problem
Thanks!