cphbusi

COPENHAGEN BUSINESS ACADEMY

DevOps part 1/4: DevOps & monitoring

Jens Egholm Pedersen
<jeep@cphbusiness.dk>

About me

Education

- 2011 - BSc. Political Science, AU, Denmark
- 2013 - ERASMUS exchange programme, EPITA Paris, France
- 2015 - BSc. Software development, ITU, Denmark
- 2018 - MSc. IT & Cognition, KU, Denmark

Experience

- 2012 – 2014 Research assistant, DemTech, Copenhagen
- 2014 - 2016 Software engineer, CERN, Switzerland
- 2016 - Assistant professor, Cphbusiness, Denmark
- 2016 - CTO, Mobilized Construction, Denmark

- Generally a nice guy
  - Don’t be afraid to ask questions

Learning how to learn

- A word on metacognition
  - What does that mean?
- Dunning-Kruger effect
  - Stupid people think they are smart
  - Why is that a bad thing?
- Continuous feedback
  - Where would you like to be when this course ends?
  - Keep evaluating yourself
    - Class participation, ask questions, search for terms you don't understand, etc.
  - … and be honest!

See also: Dunning-Kruger effect, Metacognition improves your grade!

About these lectures 1/2

- Lectures: Practical part
  - Giving you hands-on experience
  - Resolving immediate problems
  - You need your computer for the practical part
- Lectures: Theoretical part
  - Answering the ‘why’ question
  - Putting things in context (do not underestimate this)
  - You have 1 (one) job
    - You learn best by writing things down. By hand!
  - You do not need your computer for the theoretical part (!)

About these lectures 2/2

- Exploit what we prepared for you
  - Bloom's Taxonomy
  - Lecture = Comprehending
  - Lecture + Preparation = Analyzing
    - Please read the literature. Please?
  - Lecture + Preparation + Exercises = Evaluating
- When studying for the exam use 'see also'
  - **Not part of the curriculum!**

See also: Something to read, Bloom's taxonomy

Goals of LSD

- Train the student to develop large-scale IT systems, where scalability is a key characteristic
- The student must have knowledge of concepts, techniques and technologies for the continuous integration and delivery of software-based systems
- The student must be able to design, implement, and maintain large distributed systems in distributed development teams

See also: Your curriculum 2017 (pdf)

cphbusiness

Goals of the DevOps part

- Give you theoretical and practical knowledge on maintening and operating large systems

1) Monitoring 2. November
2) Logging 9. November
3) Scaling 16. November
4) Security 23. November

- Essentially everything that happens *around* the code

See also: Your curriculum 2017 (pdf)

Goals for today

- Understand what DevOps is and why it's needed
- Understand what SLAs are and why they're needed
- Understand what monitoring is and why it's important
- Gain practical knowledge on how to use and install monitoring software for your project
- Trade projects!

Literature: DevOps introduction

Toyota factory example

- You have a factory manufacturing cars
- Each car requires several parts
- Each part is manufactured at a station
- Each part depends on one or more parts

- What are important factors to optimize production?

See also: DevOps by example, Toyota line layout

Toyota factory example

- You have a factory manufacturing cars

- Rank these items in terms of how wasteful they are:
  - One station works faster than the surrounding stations
  - One station works slower than the surrounding stations
  - One station breaks down and needs to be repaired
  - A fire breaks out and the entire factory closes
  - All stations needs to shut down for upgrades one hour

See also: DevOps by example, Toyota line layout

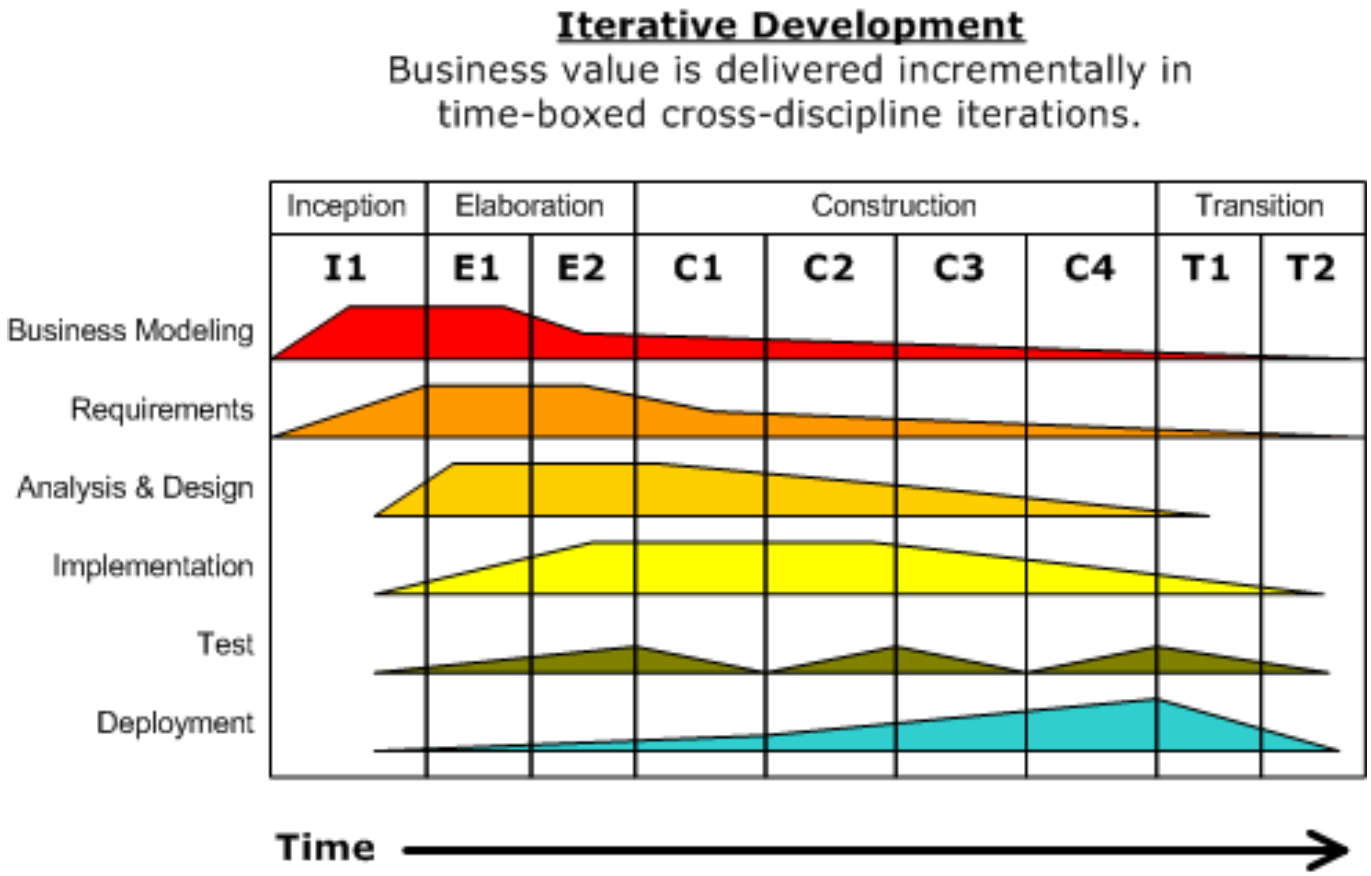
A long time ago...

- Sommerville 2011: Sofware Engineering, 9th edition
- 800 pages of software engineering processes

- 15 pages on software maintenance and legacy systems

- "Software maintenance is the general process of changing a system after it has been delivered"
  - What's good and what's bad about this statement?

Iterative development



… and then what?!

After release

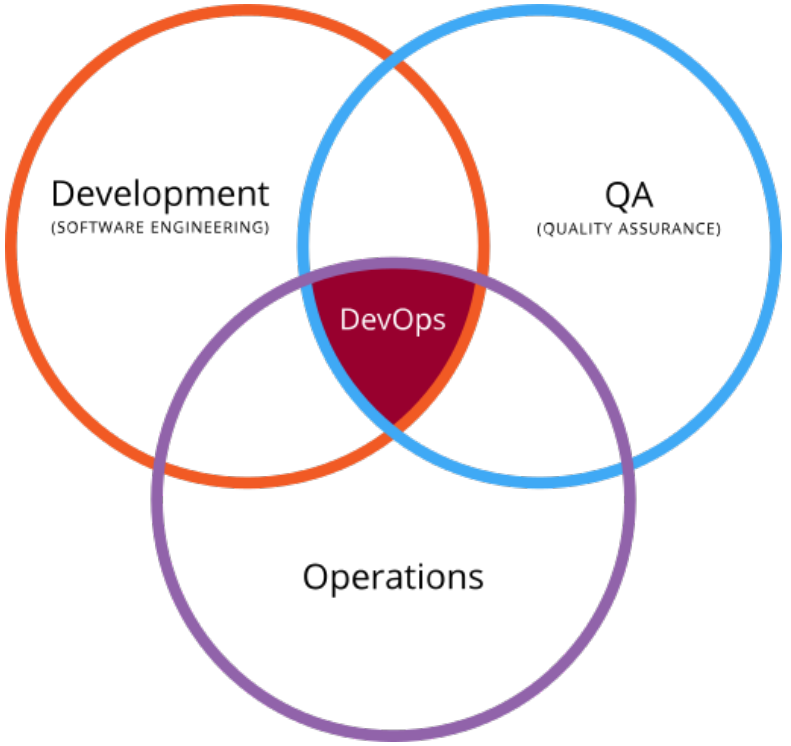
- What happens after you release?
  - What happens if you need to upgrade the system?
  - What happens if your system goes down?
  - What happens if your system is flooded by traffic?

- How can we use the car manufacturing example?

What is DevOps?

- Formally a combination of “Development” and “Operations”

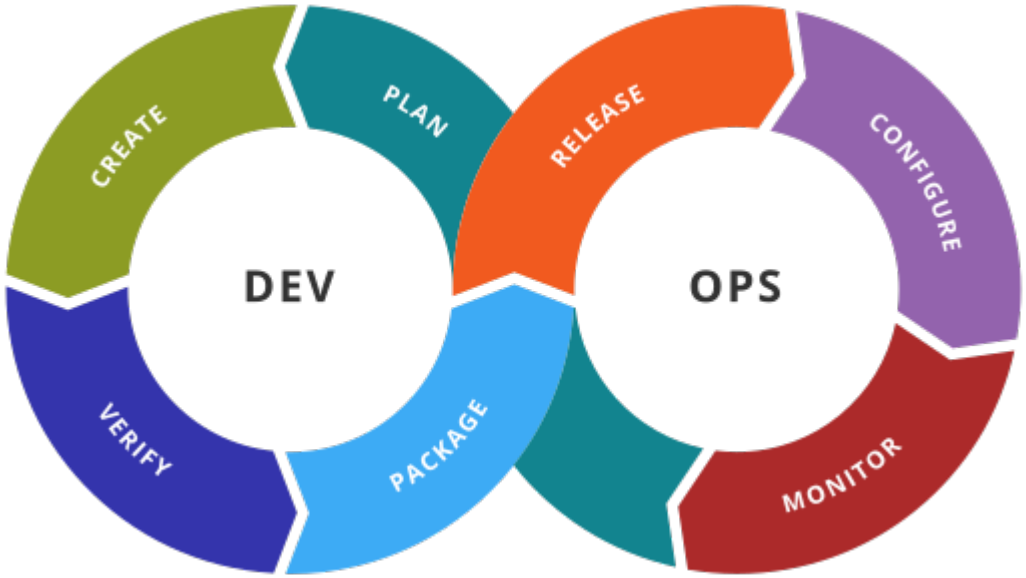
- Main ideas of DevOps
  - Automation
  - Monitoring



See also: DevOps on Wikipedia

So what *is* DevOps?

- Still young: thousands of interpretations (and titles)
- Not a clear impact (measurable in 1/5 organisations)
- My take on this: simply planning ahead



See also: DevOps on Wikipedia

cphbusiness

Benefits of ~~planning~~ DevOps

1) Faster time to market
2) Improved deployment frequency
3) Lower failure of new releases
4) Faster fixes
5) Faster recovery

See also: DevOps on Wikipedia

Recap: DevOps

- What is DevOps?
  - Tools and techniques to keep the software operational
- Why do we need it?
  - To prepare for the future; Murphy's law
- What will it give us?
  - Faster time to market
  - Fewer failures (and grey hair)
  - Happier customers and developers
- "DevOps is just the agile principle, taken to the full enterprise" - Tony Bradley, TechPerspective

See also: Murphy's law, 10 ways DevOps Is changing enterprise IT

cphbusiness

Service-level agreements (SLA)

- “An official commitment that prevails between a service provider and a client” - Wikipedia
- A measurement of good service

- Why do we need it?
  - Because we need proof!

See also: SLA on Wikipedia

SLA metrics

- How would you measure good service?
  - Think about your ISP. How would you measure them?
  - What about your cashier in Fakta?

- Main point #1: it depends what service you offer
- Main point #2: Common metrics (for web)
  - Uptime/availability (usually percentage of all time)
  - Mean response time (average time before answer)
  - Mean time to recover (time to recover after outage)
  - Failure frequency (number of failures/timeouts over time)

See also: SLA on Wikipedia, classification of SLA metrics

cphbusiness

~~Toyota~~ software factory example

- You have a factory manufacturing cars
- You just signed an SLA
- How do you know if you live up to it?!

See also: DevOps by example, Toyota line layout

Monitoring

- What is monitoring to you?

- Passive monitoring
  - Detecting problems after they occur
  - Example: Dashboards
- Active monitoring
  - Detects problem while they occur
  - Example: Alarms

See also: Active vs passive web performance monitoring

Passive vs. active monitoring

- Passive versus active monitoring

- Common SLA metrics (for web)
  - Uptime/availability (usually percentage of all time)
  - Mean response time (average time before answer)
  - Mean time to recover (time to recover after outage)
  - Failure frequency (number of failures/timeouts over time)

See also: Event monitoring on Wikipedia

Monitoring in practice

- Passive monitoring
  - How would you do it?
  - Grafana
  - Alternatives: Kibana

- How to install Grafana
  - `docker run`

```
--net=”host”
–-name grafana
-p 3000:3000
-d grafana/grafana:4.5.2
```

- Next step: Test application

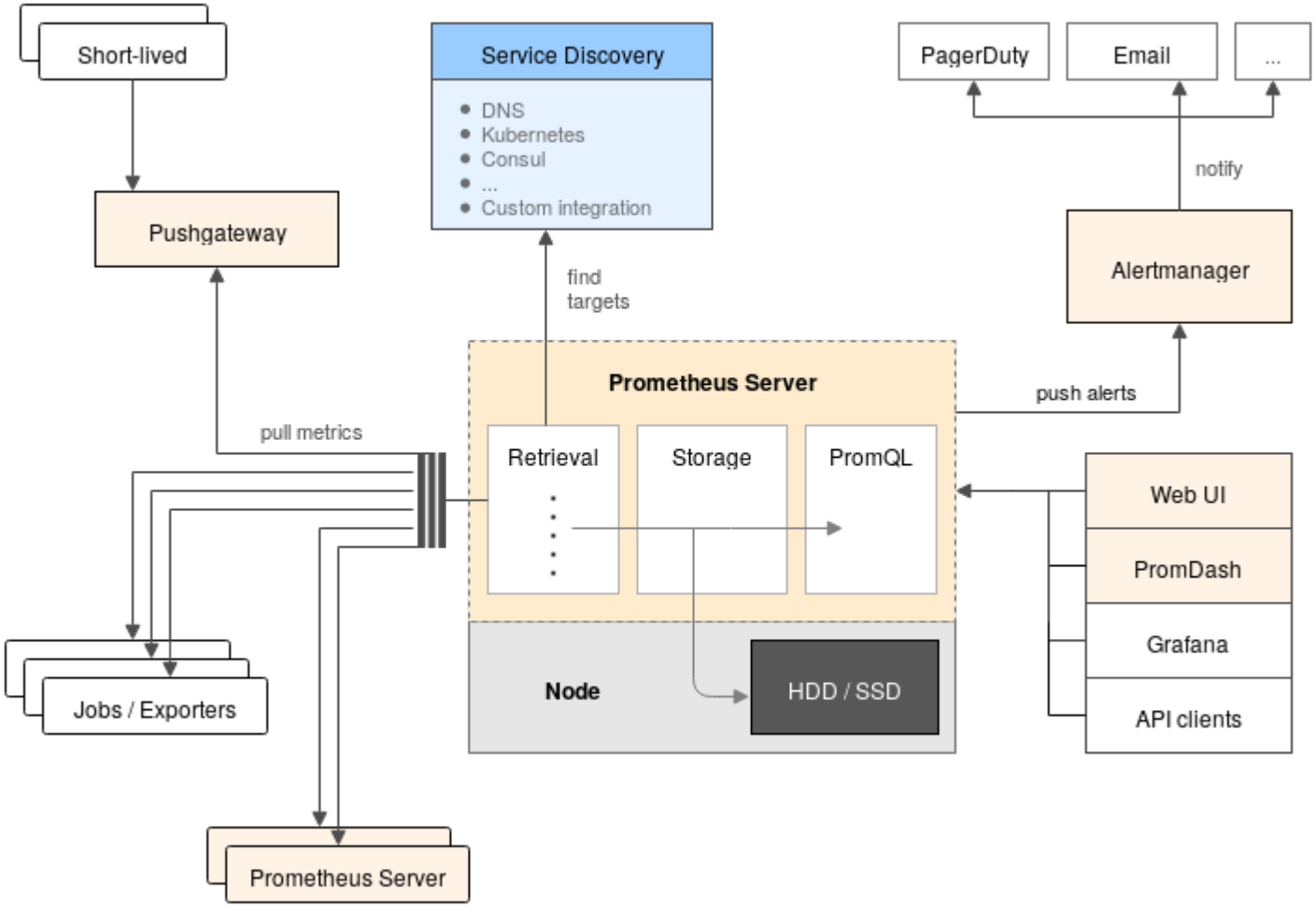
See also: Demo dashboards

Prometheus

- Basically a time series key-value storage
- Either pulling or pushing data
  - We will only use data pull
- Targets (for pulling) via service discovery or config

See also: Prometheus documentation introduction

About Prometheus



Setting up a demo service

- To present a webservice to prometheus, we need to expose Prometheus metrics at `/metrics`
- My custom example in nodejs
  - `docker run`
    `-p 8080:8080`
    `-d jegp/nodejs-prometheus`
- Open localhost at :8080

cphbusiness

Monitoring in practice

- Monitoring backend
  - Prometheus
  - Alternatives: Elasticsearch

- Installing Prometheus
  - `docker run`

```
--name prometheus
--net=”host”
-v `pwd`/prometheus.yml:/etc/prometheus/prometheus.yml
-p 9090:9090
-d prom/prometheus
```

See also: Prometheus on docker hub

What just happened?

1) We installed Grafana on `:3000`
2) We installed an app exposing `:8080/metrics`
3) We installed Prometheus on `:9090`
   - Prometheus is configured with `prometheus.yml`

cphbusiness

Coming full circle

1)Open Grafana, login with admin/admin

2)Add Prometheus as a data source

3)Open a dashboard and behold (!)

A note on Prometheus syntax

- Grafana can connect to multiple backends
- When querying Prometheus you need to use their syntax
  - `http_requests_total`
  - `http_requests_total{job=”nodejs-prometheus”}`
  - `rate(http_requests_total[5m])`
  - `sum(rate(http_requests_total[5m])) by (job)`

See also: Prometheus queries

cphbusiness

Prometheus dashboards

- Just because you can include everything doesn't mean you should!

- `https://grafana.com/dashboards`

See also: Make DevOps Dashboard tell a Story

Alerts

- Active monitoring
  - Get notified when something goes wrong

- Alerts in Grafana:
  - On dasboards
  - Or via channel
    - Mail, PagerDuty, Slack, Telegram

Recap

- Service-level agreement (SLA)
- Monitoring
  - Why do we need it?
- Pratical:
  - Installation of Prometheus and Grafana
  - Dashboarding
  - Alerts

cphbusiness

System hand-over

- You'll take over a system from another group
- You'll not code on that system
- You'll alert the owners when something needs fixing

- Your jobs right now
  - Establish a clear communication channel between groups
  - Hand-over the hand-over document for the hand-over
  - Hand-in the hand-over on the moodle hand-in, with a link to the hand-over

cphbusiness

Next hand-in: Monitoring

- Make an SLA with the group operating your system
  - Can be as sofisticated as you want it to be, but must include an uptime requirement
- Setup Grafana and Prometheus
  - Install Grafana and Prometheus on a globally accessible server
  - Alter your services to expose Prometheus metrics
  - Configure Prometheus to monitor your services
    - All of them!
- Setup a dashboard that displays:
  - The metrics necessary to uphold the SLA
  - A text element (on the dashboard itself) linking to or including the SLA
- Hand-in:
  - A link to your Grafana dashboard with your metrics and SLA