

software engineering in practice

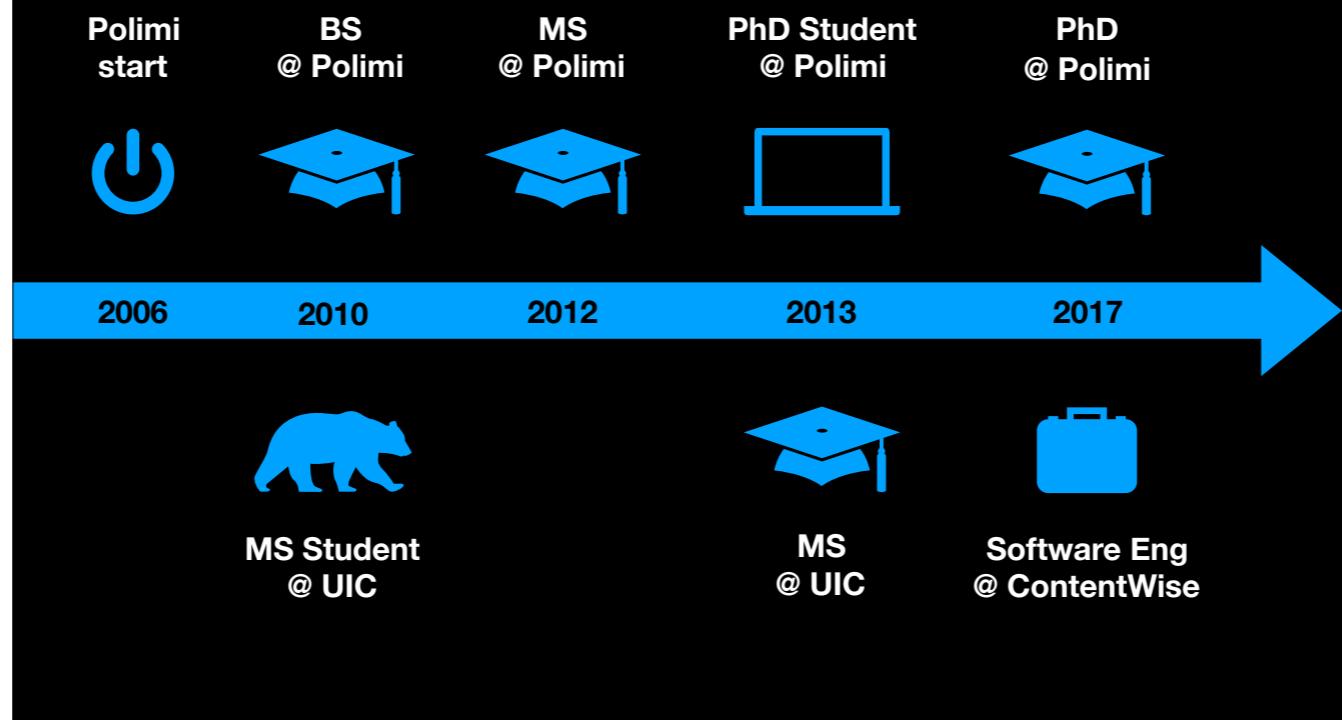
experience from a former Polimi student

Marco Miglierina

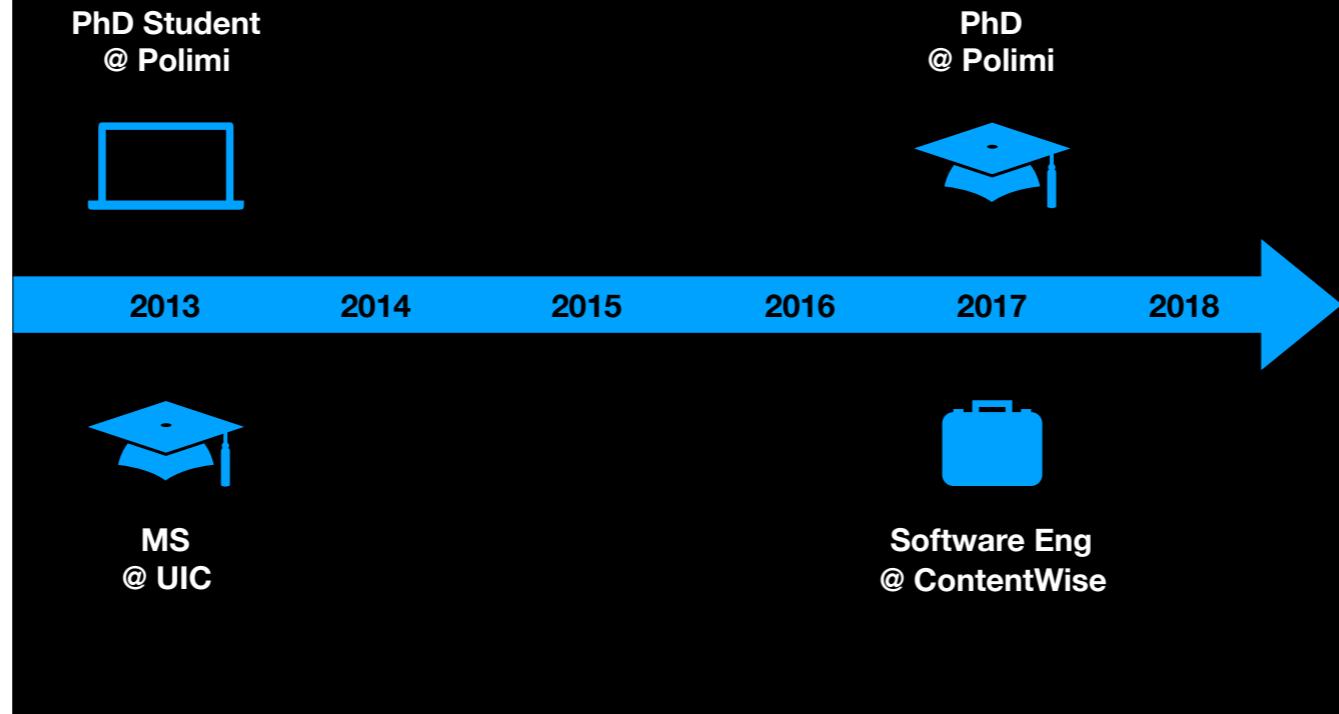
Overview

- About me
- Student life @ Polimi / UIC
- Software engineer @ ContentWise

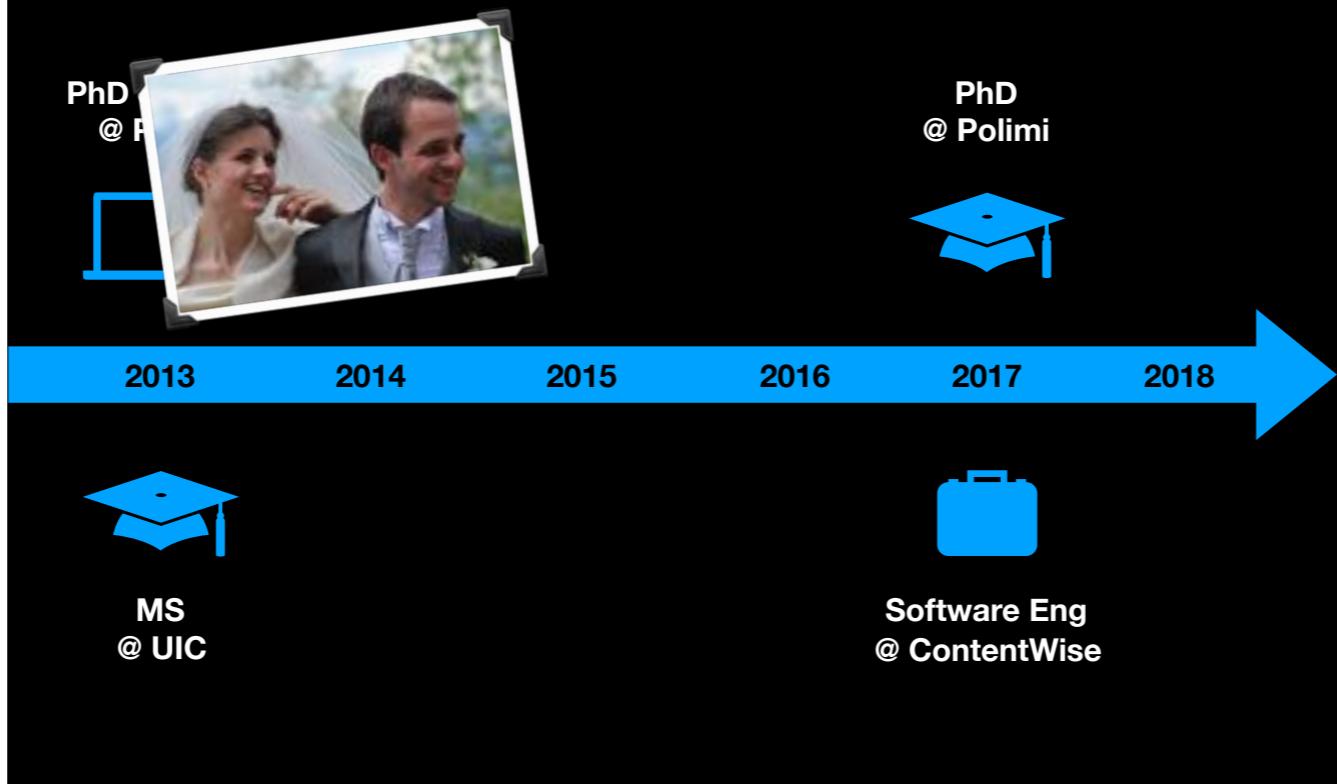
About me



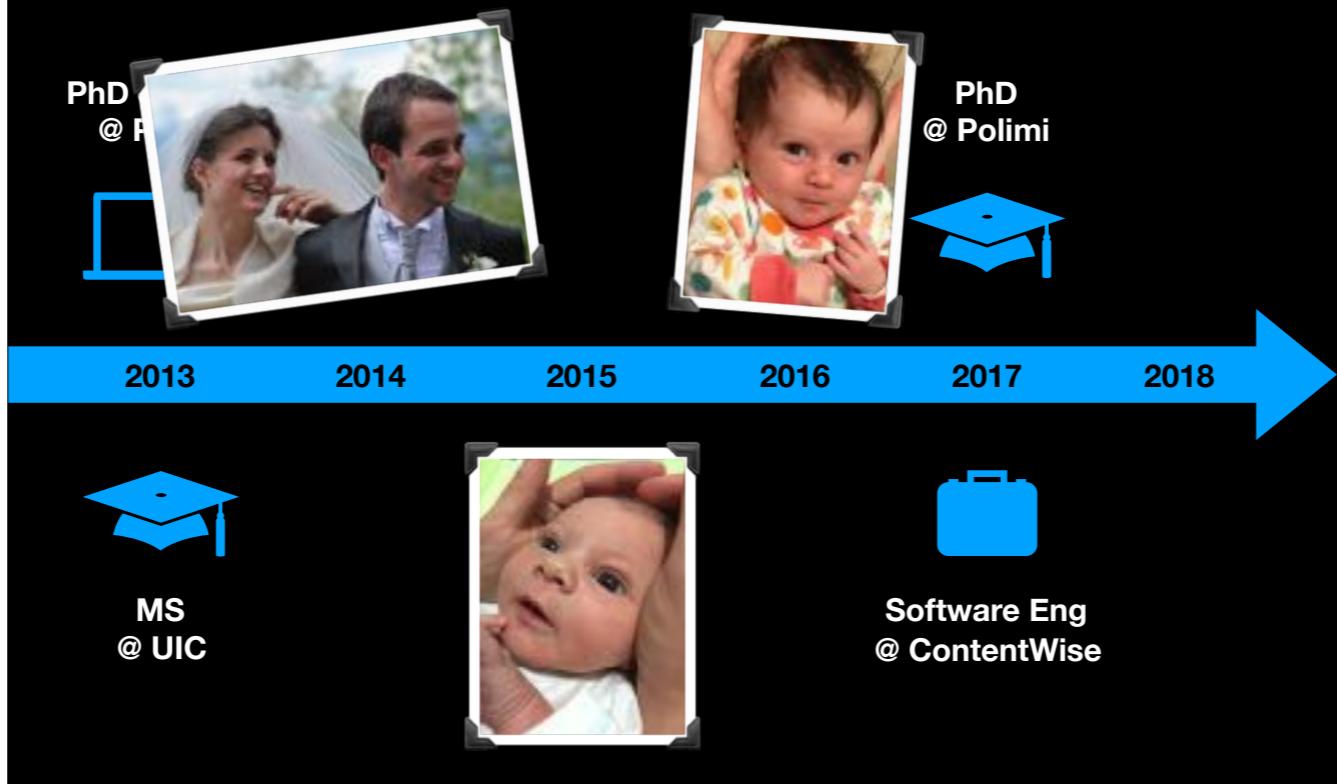
About me



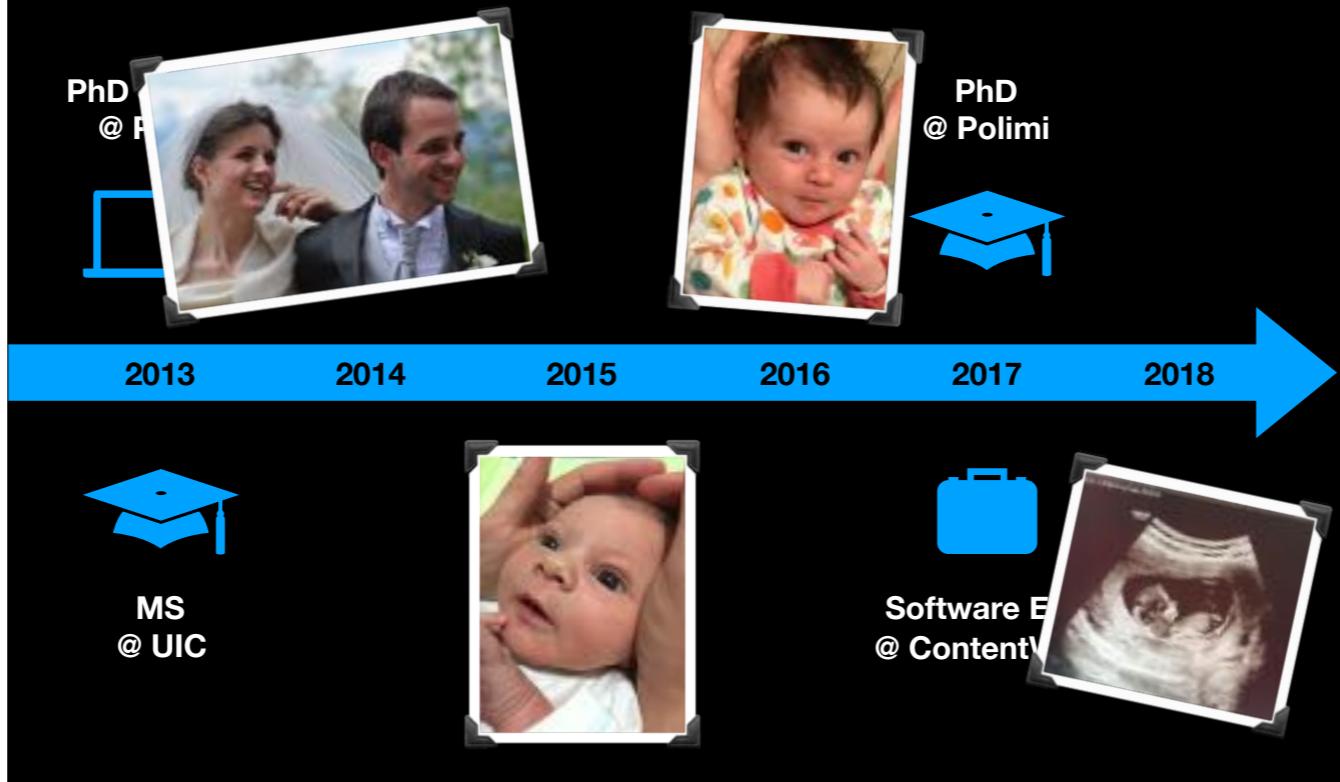
About me



About me



About me



life @ Polimi



ingenuity

- Why computer science?

ingenuity

- Why computer science?
- Because I love computers!



reality

reality

Communication Networks
and Internet

Algorithms

Operating Systems

Databases

Computer Architectures

Electronics

Fundamentals of Computer Science

Software Engineering

Maths

Physics

Theoretical Computer Science

Information Systems

Statistics

Probability

Circuit Theory

Automatic Control

reality

Communication Networks
and Internet

Electronics

Information Systems

Fundamentals of Computer Science

Statistics

Algorithms

Probability

Operating Systems

Circuit Theory

Maths

Databases

Software Engineering

Physics

Computer Architectures

Control Theory

Theoretical Computer Science

reality

Communication Networks
and Internet

Electronics

Information Systems

Algorithms

Fundamentals of Computer Science

Statistics

Operating Systems



Probability

Maths

Circuit Theory

Databases

Software Engineering

Physics

Control Theory

Computer Architectures

Theoretical Computer Science

first attempts

Computer Architectures + Physics



=

Definition of an FPGA based on Quantum Dot Cellular Automata and Realisation of Development Tools



after 3 years I had no idea of what I liked the most, where I wanted to invest more...

I did a thesis where I mixed up together some of my passions: computers and physics... with a pinch of science fiction...

It was fun, interesting, but not my thing

I believe I can fly



opportunity



Chicago?!?!!!



those were actually my parent's money, which I need to thank for the opportunity they gave me... not everyone can afford it.



those were actually my parent's money, which I need to thank for the opportunity they gave me... not everyone can afford it.

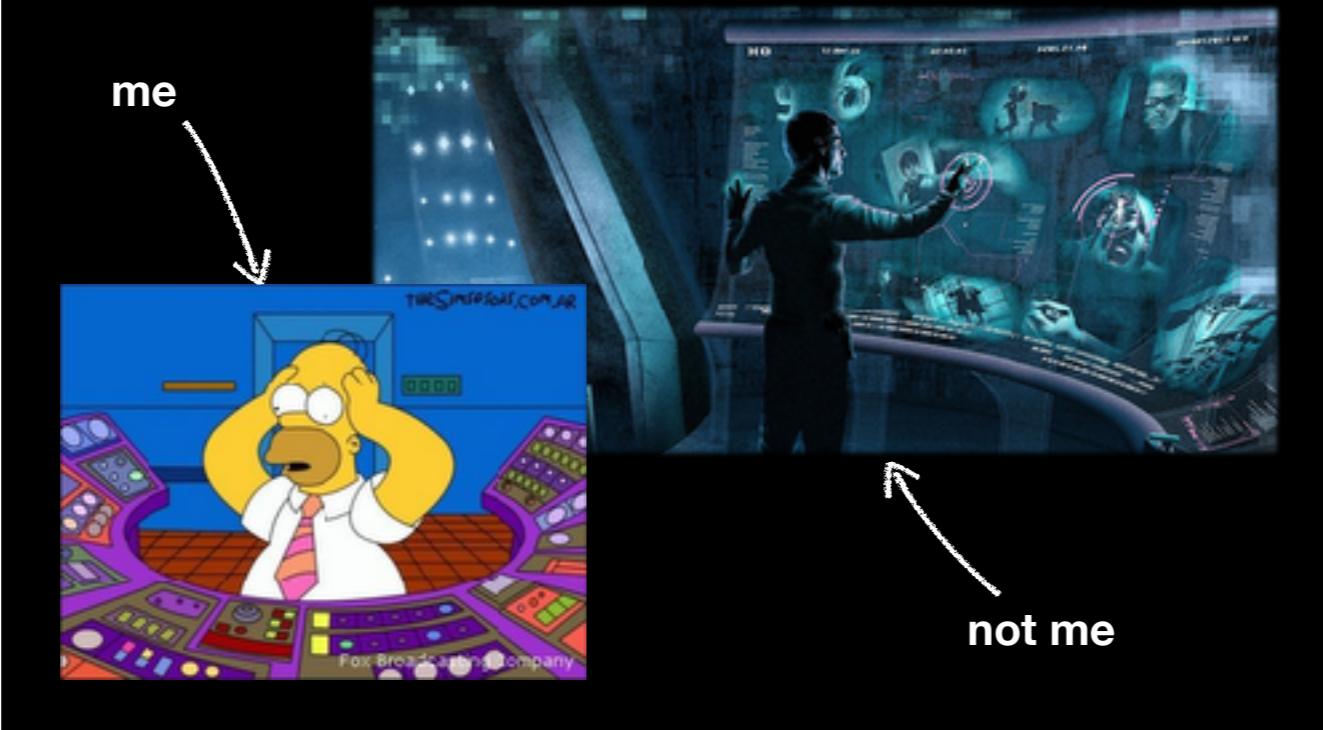
reality... again



I was super excited after the graduation and I was so looking forward to leaving for Chicago. I started the UIC program, the first semester here at Polimi was very hard, mix between the Polimi and US way.

Polimi: high standard, high requirement + US mid term, finals and project, no retries.

reality... again



Also, I met brilliant students, smarter than me, that were coming with me to Chicago and they knew so much more than me about computers... and I felt a little bit like...



...Jon Snow...?

I'm not saying reality is bad, reality is just fantastic to me (I would never give life to my children otherwise). However, we start dreaming and detach from reality very easily. Luckily, when we do, reality comes back to us and awakens us.

It was a great opportunity for me, first to become a little more humble, and then to realise there is still so much to know and discover!



...Jon Snow...?

I'm not saying reality is bad, reality is just fantastic to me (I would never give life to my children otherwise). However, we start dreaming and detach from reality very easily. Luckily, when we do, reality comes back to us and awakens us.

It was a great opportunity for me, first to become a little more humble, and then to realise there is still so much to know and discover!

Life @ UIC



classes I took

Artificial Intelligence

Neural Networks



Expert Systems

Data and Web Semantics

I studied some of AI, neural network, expert systems and semantic web

Awesome experience, but it was hard, night work projects.. several projects.

Just because we are used to high level, in fact, we (Polimi students) are known to be very good students there and we rocked.

finals week



Last week, the week of finals it's an all-in.. in one week 4 exams and 4 project presentation... I was not even able to understand what it was going on. The last day I had the last exam and the last project presentation. Me, and other students, did not even sleep... did everything, don't remember much... except an awesome basket ball game that night Bulls vs Lakers.

towards clouds and automation

- “Advanced topics in software engineering” class project
 - *Benchmarks for Cloud Deployed Web applications*
- master thesis
 - *Model based control for multi-cloud applications*

towards clouds and automation

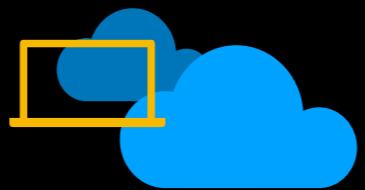
- “Advanced topics in software engineering” class project
 - *Benchmarks for Cloud Deployed Web applications*
- master thesis
 - *Model based control for multi-cloud applications*



what next?



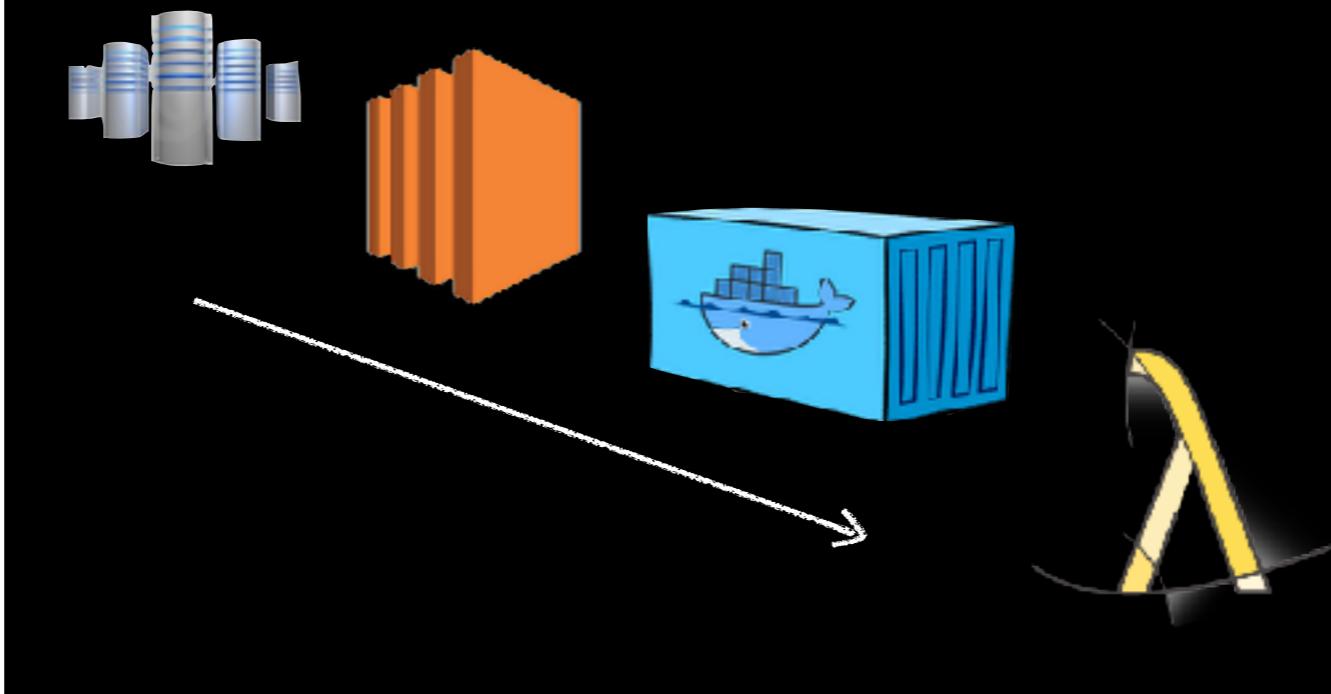
PhD



my phd thesis

***Monitoring Modern Distributed Software
Applications: Challenges And Solutions***

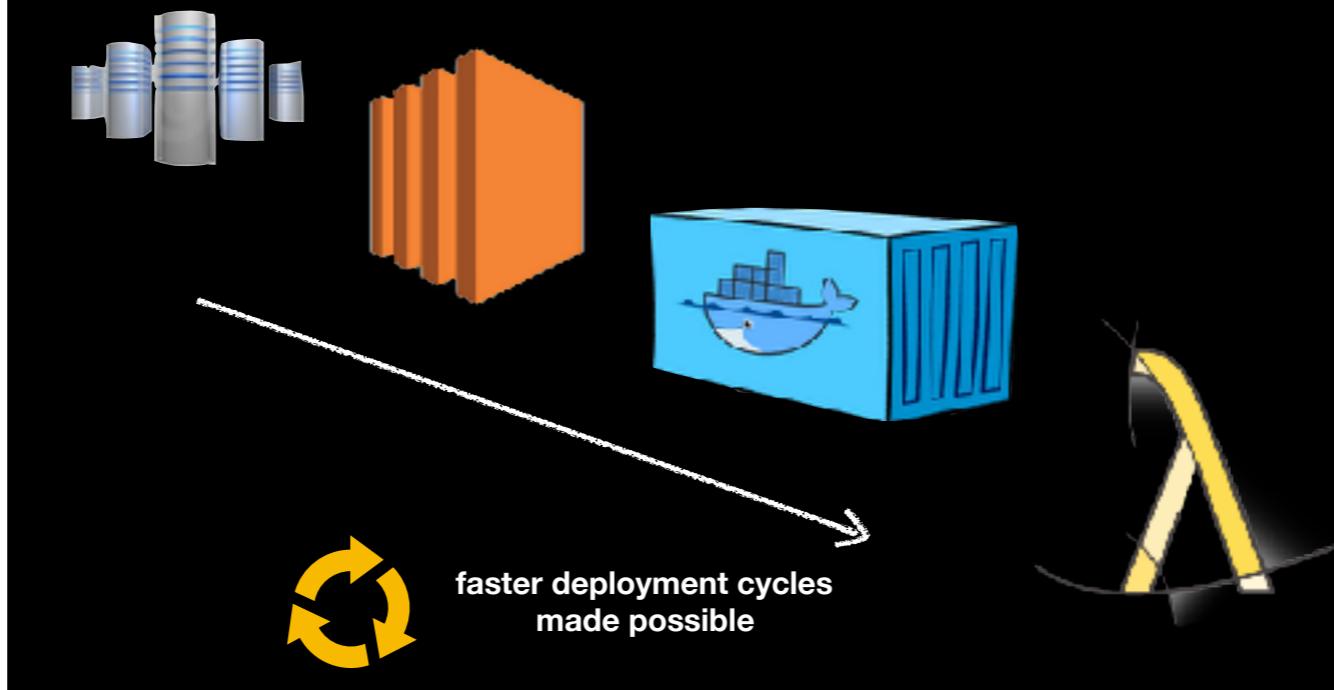
paradigm shift



cloud computing brought a paradigm shift, no upfront investments, machines, network infrastructures, deployed in minutes...

So it became much simpler to deploy new versions, and test them... you have a new version, you deploy a new environment, you test it, you route traffic to it.. Since deployment cycles became faster, you could not rely on the old monolithic apps... so microservices, a single application deployed as a set a suite of small services.. independent cycles, independently deployable (Fowler)

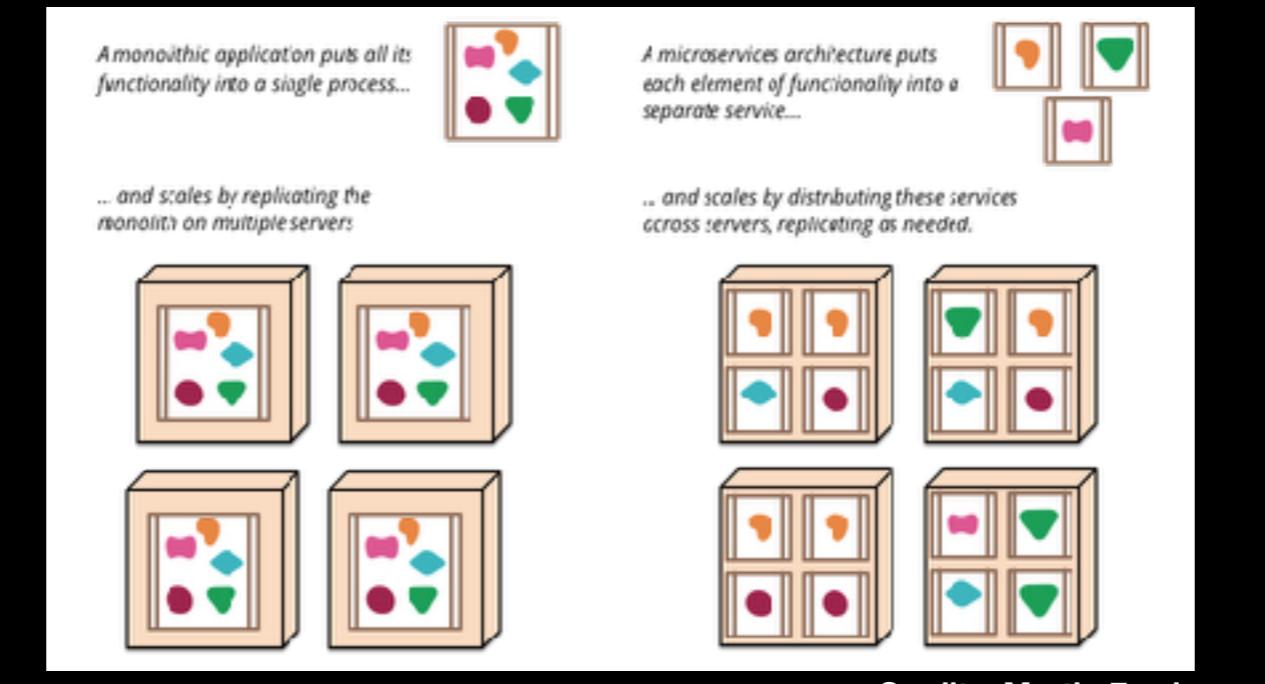
paradigm shift



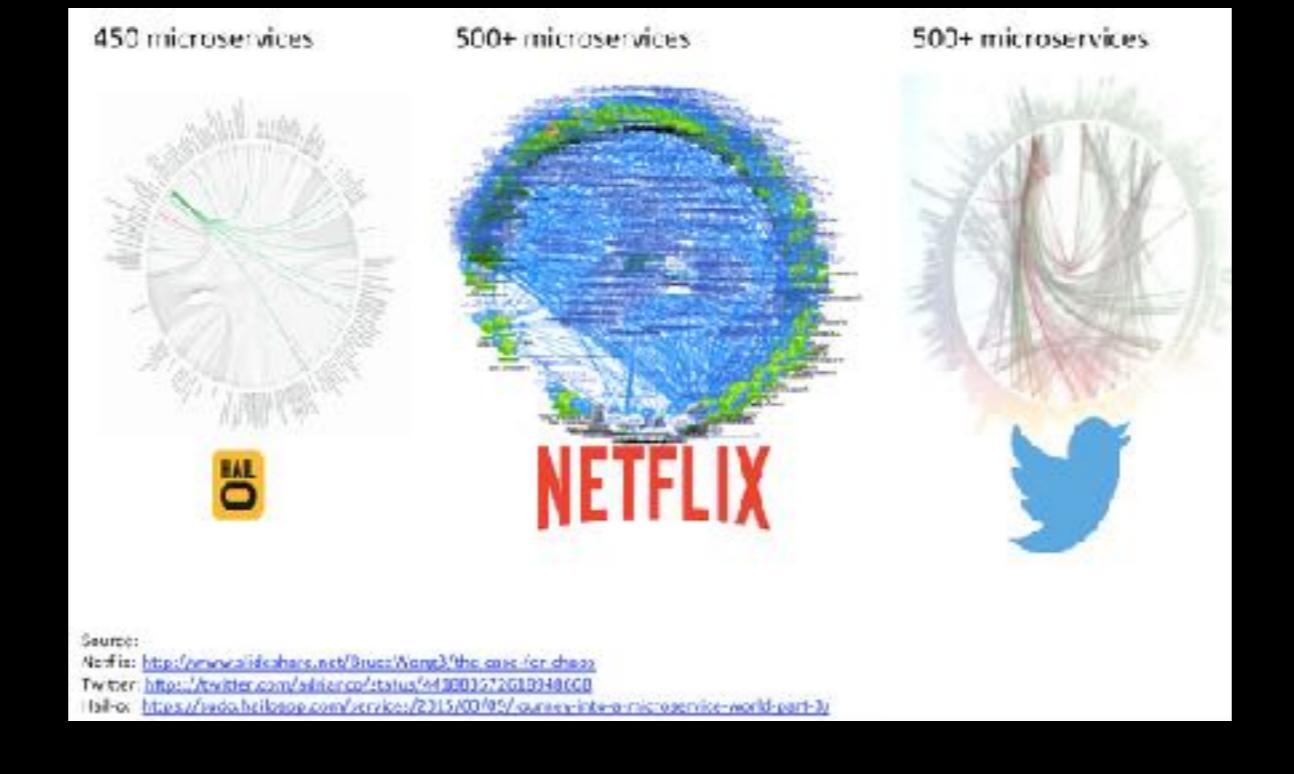
cloud computing brought a paradigm shift, no upfront investments, machines, network infrastructures, deployed in minutes...

So it became much simpler to deploy new versions, and test them... you have a new version, you deploy a new environment, you test it, you route traffic to it.. Since deployment cycles became faster, you could not rely on the old monolithic apps... so microservices, a single application deployed as a set a suite of small services.. independent cycles, independently deployable (Fowler)

from monoliths to microservices



death stars



Some examples of death stars (microservices networks)

monitoring challenges

- Volatile and dynamic resources
- heterogeneous resources
- huge amount of monitoring data
- distributed metrics ownership

- volatile and dynamic environments <-> self discovery
- heterogenous resources <-> abstraction models
- what metrics are actually important <-> qos-driven
- metrics come from multiple services often managed by different teams <-> standardization

solutions

- Volatile and dynamic resources => **automated discovery**
- heterogeneous resources => **abstraction models**
- huge amount of monitoring data => **qos-driven analysis**
- distributed metrics ownership => **standardization**

- volatile and dynamic environments <-> self discovery
- heterogenous resources <-> abstraction models
- what metrics are actually important <-> qos-driven
- metrics come from multiple services often managed by different teams <-> standardization

phd challenges

- responsibility
- study what others are doing
- find new and unsolved problems
- time management
- support your findings
- public talks

I can't say PhD was easy for me... PhD was hard (Polimi in general isn't easy, I think you know)... However, PhD is not just you studying and solving problems professor's give to you. It's about you becoming more independent, finding problems, read what other researchers are doing, understanding how to manage your time. And this last one is probably one of my worse problem.

Also, I'm that kind of guy who gives his best when is behind a computer, doing nerdy stuff, certainly not the kind of guy who can stand in front of tens of people talking...

but here I am



however here I am somehow... in front of you talking about my experience.

I had the opportunity to travel and meet a lot of brilliant people both from the industry and from the academia. Passionate about what they are studying and made me even more eager to learn.

Whatever I was asked to do, to teach, every course I took, it made me do a little step towards finding my place, what I like to do.

And for this I need to thank my advisor, Prof Di Nitto, who was always able to value my work, especially when I was not satisfied.

I'm starting to understand, and I'm sure time will make me even more certain, the meaning of all those years.

next step



now I have a different curriculum
companies have probably changed

I know what kind of work I want to do (at least to start with... then.. who knows)

I discovered my passion for automation in software development and everything that can speedup development workflow
looking for a product oriented company for a position as



the product



Basically, we recommend what movies you may like to watch

CS Category: recommending systems

A platform enabling our customers to offer personalised content.

Our first customer was sky, now we have ... all around the world.

the younger brother



Knowledge factory:
semi-automated metadata enrichment
newer, microservices, docker containers, spring and scala

we are expanding towards different domains

How to respond to increasing demand?

- Agile: scrum
- Git flow
- Continuous integration
- Automated deployments
- DevOps
- Infrastructure as code
- Continuous Delivery
- Chaos engineering

Manifesto for Agile Software Development



We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

<http://agilemanifesto.org>

2001, software practitioners gathered sharing their experience and why so many software project were failing. 4 Main values. Values in the right are still important, but values on the left are more.

scrum



product



stakeholders

There are different agile frameworks that try to “implement” the agile manifesto. And each team would adapt it to their needs and requirements. So I will talk about what we adopted in ContentWise: Scrum.

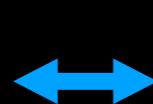
scrum



product



stakeholders



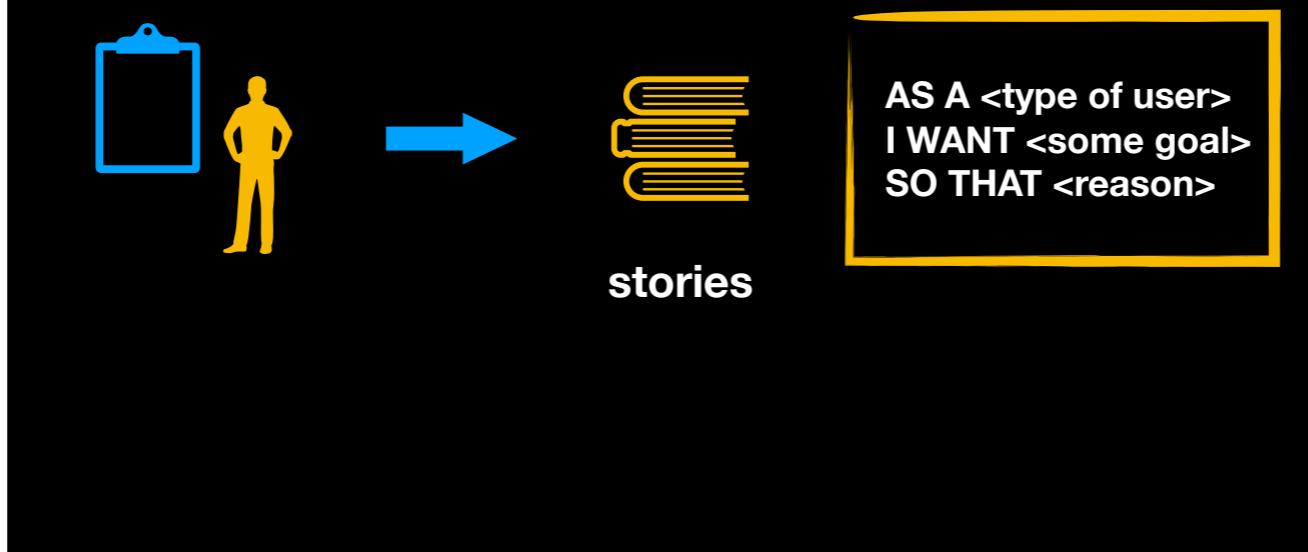
product owner

scrum



requirements

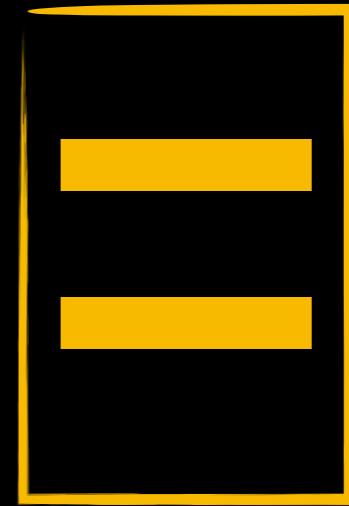
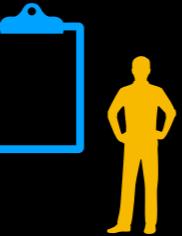
scrum



Examples:

- AS A ContentWise Operator I WANT the search box to autocomplete while I type SO THAT I can search faster among existing recommending algorithms
- AS UI developer I WANT administration REST API to return a user friendly message when an error occurs SO THAT that can be shown to the user

scrum



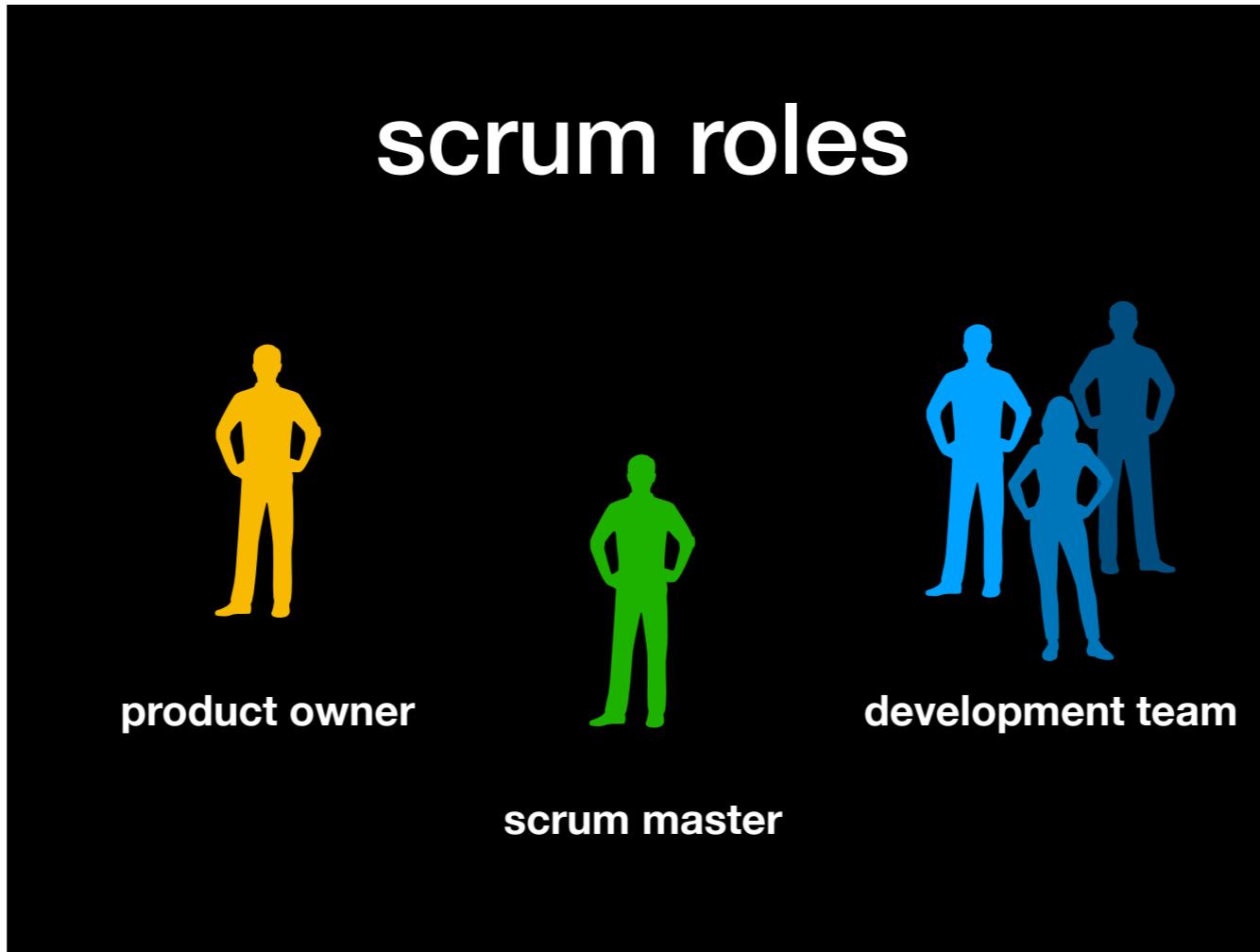
backlog

scrum



backlog

scrum roles



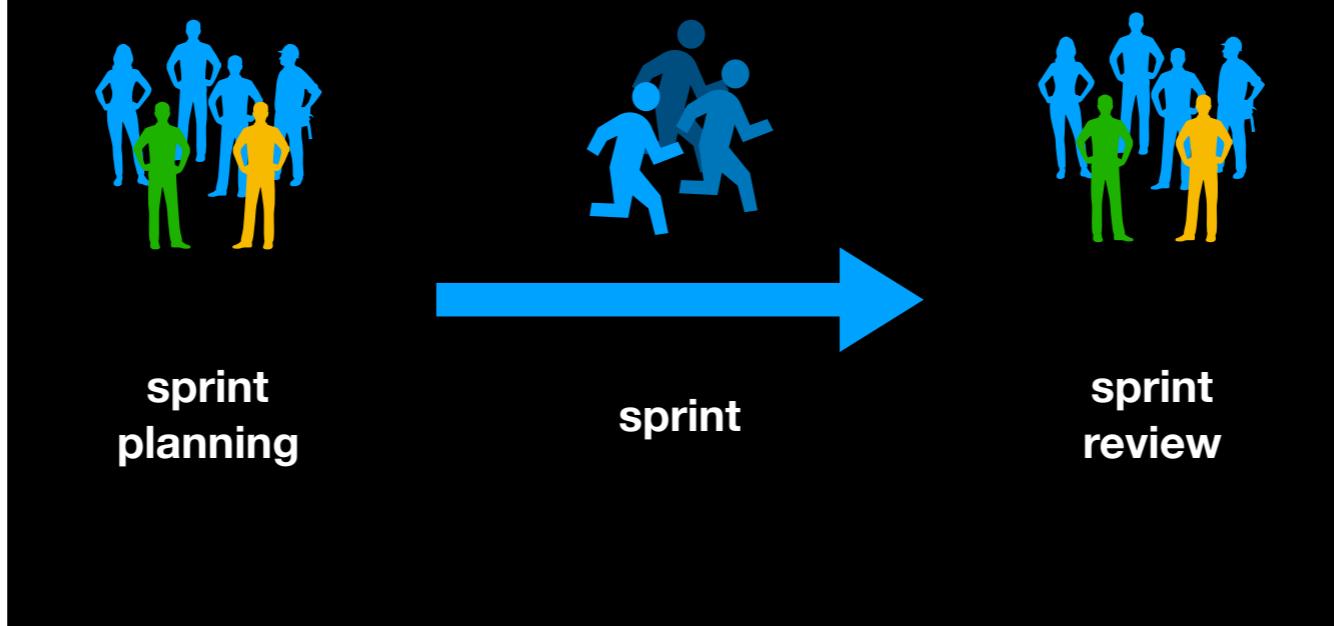
scrum master: responsabile del team, che le storie vengano portate a termine, che il team possa focalizzarsi sulle storie, fa da cuscinetto verso qualsiasi distrazione

scrum master: ensures that the scrum framework is followed and is the buffer between the team and any distracting influence, team coaching, help understanding when a story is done

development team: responsible of building product increments, 3 to 9 members, cross functional and self-organizing... obviously there can't be a perfectly homogeneously distributed team (utopia). Anyway, there shouldn't be single point of failures, or bottlenecks.

Also, not just programming: documentation, design, spikes, backend/frontend development, automation, testing..

scrum workflow



sprint planning

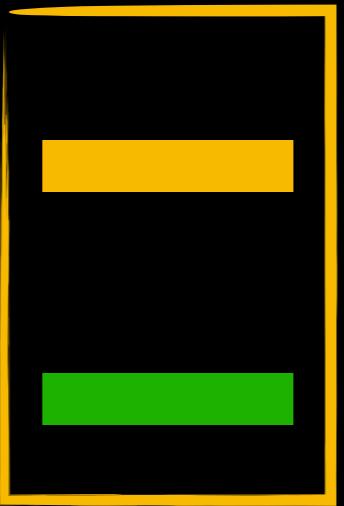


backlog



sprint 48

sprint planning



backlog



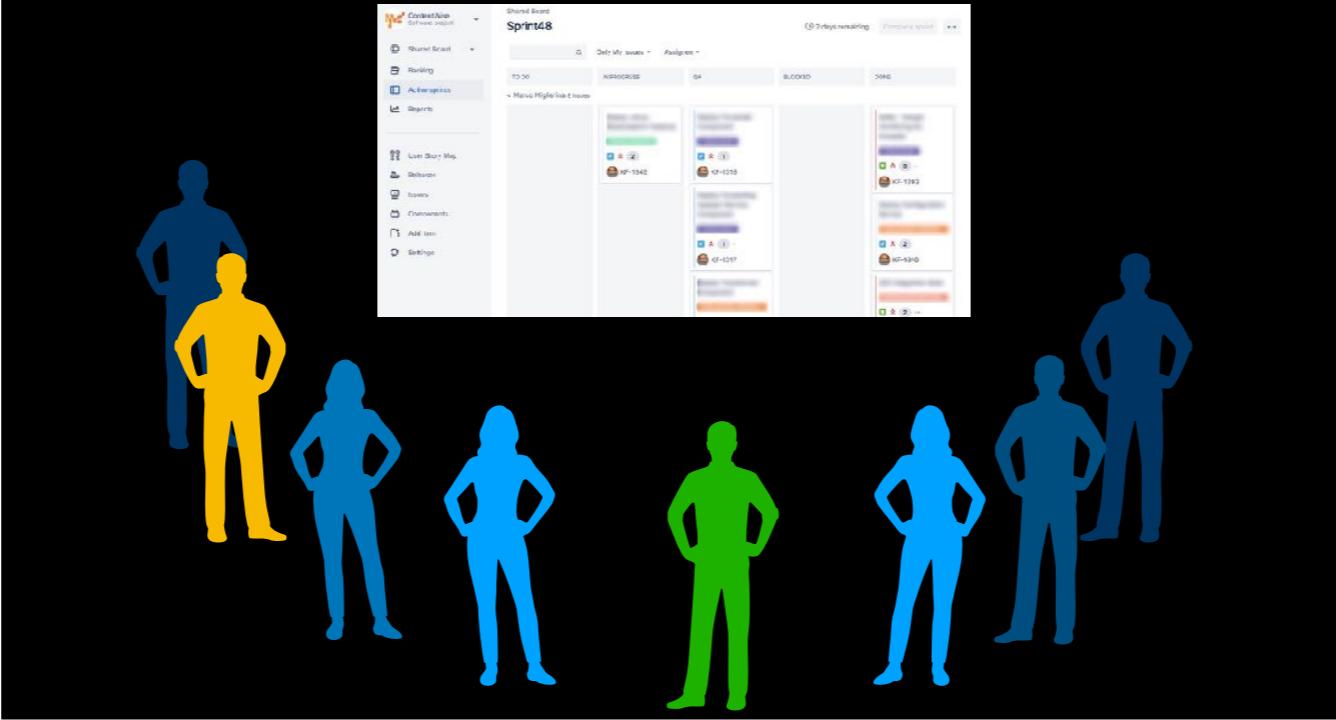
sprint 48

the scrum board

The screenshot shows a Scrum board titled "Sprint48". The board is divided into five columns: TO DO, IN PROGRESS, QA, BLOCKED, and DONE. Each column contains several stories, each with a small icon, a story ID (e.g., KF-1048, KF-1015, KF-1017, KF-1010, KF-1092), and a progress bar indicating completion status.

Column	Story 1	Story 2	Story 3	Story 4	Story 5
TO DO	KF-1048	KF-1015	KF-1017	KF-1010	KF-1092
IN PROGRESS	Progress: ~				
QA	Progress: ~				
BLOCKED	Progress: ~				
DONE	Progress: ~				

daily stand-up



scrum pros

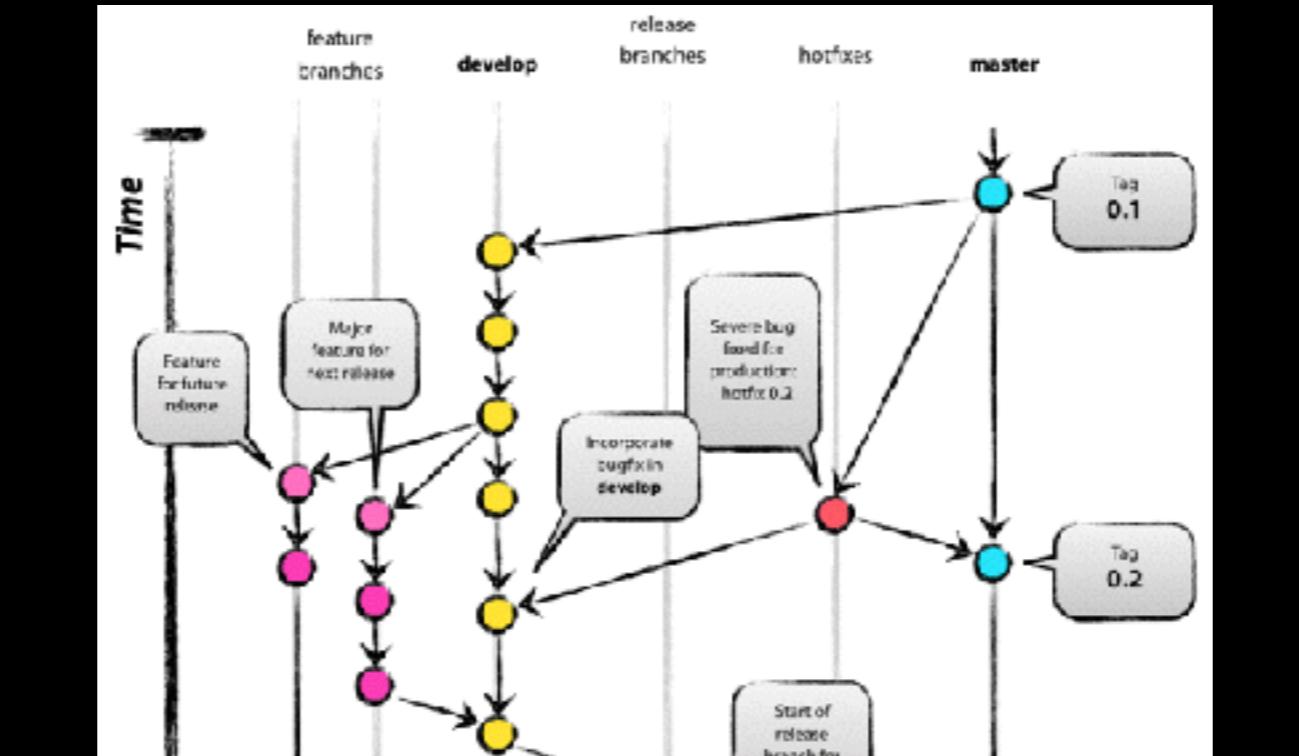
- Features are split into small units of work:
 - predictable effort
 - dev focus on one problem at a time
 - work progress is observable



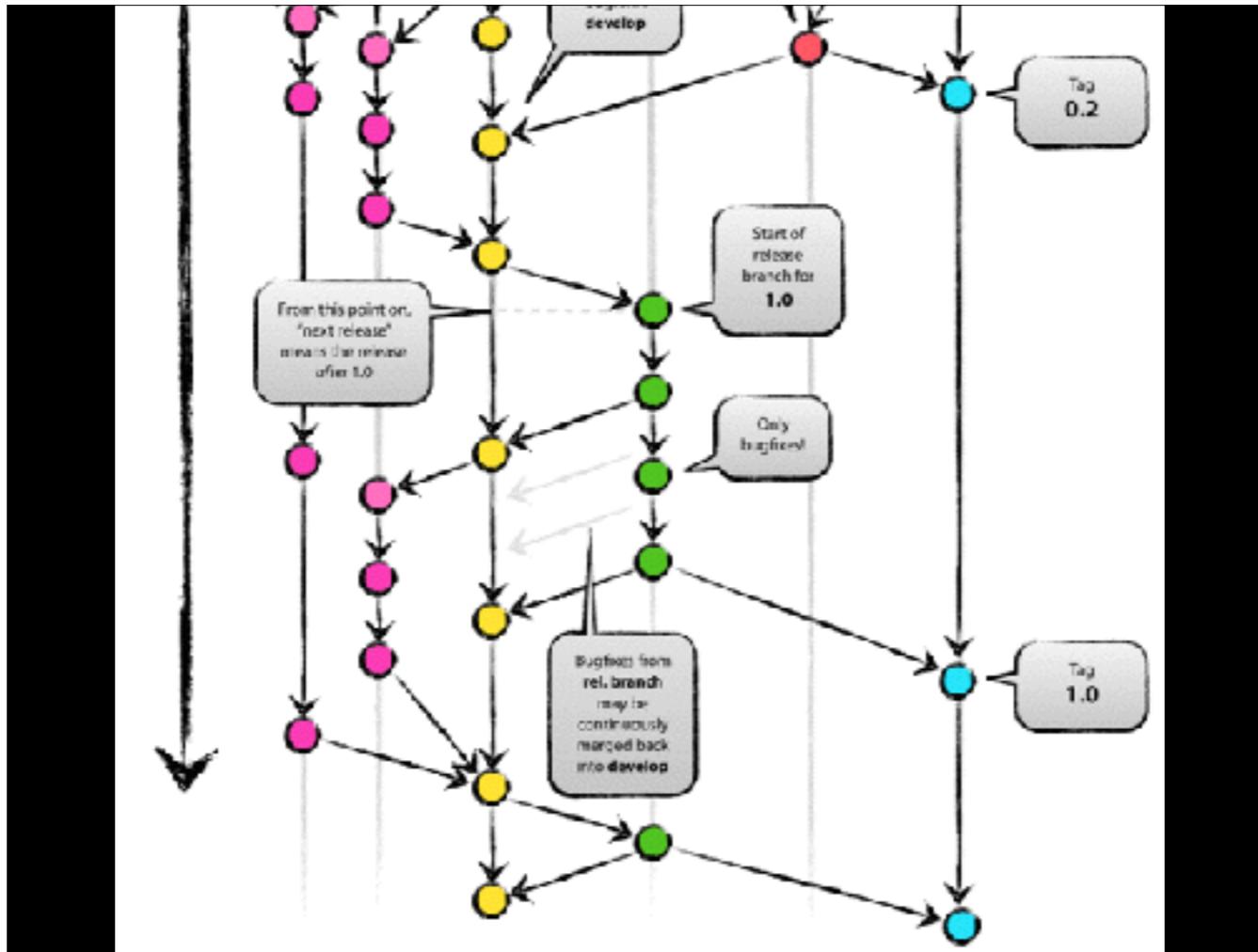
scrum cons

- easy to focus only on the framework and speed
- requires skilled and cross-functional team
- stories often lack detailed specification (they are often just placeholders)

git flow

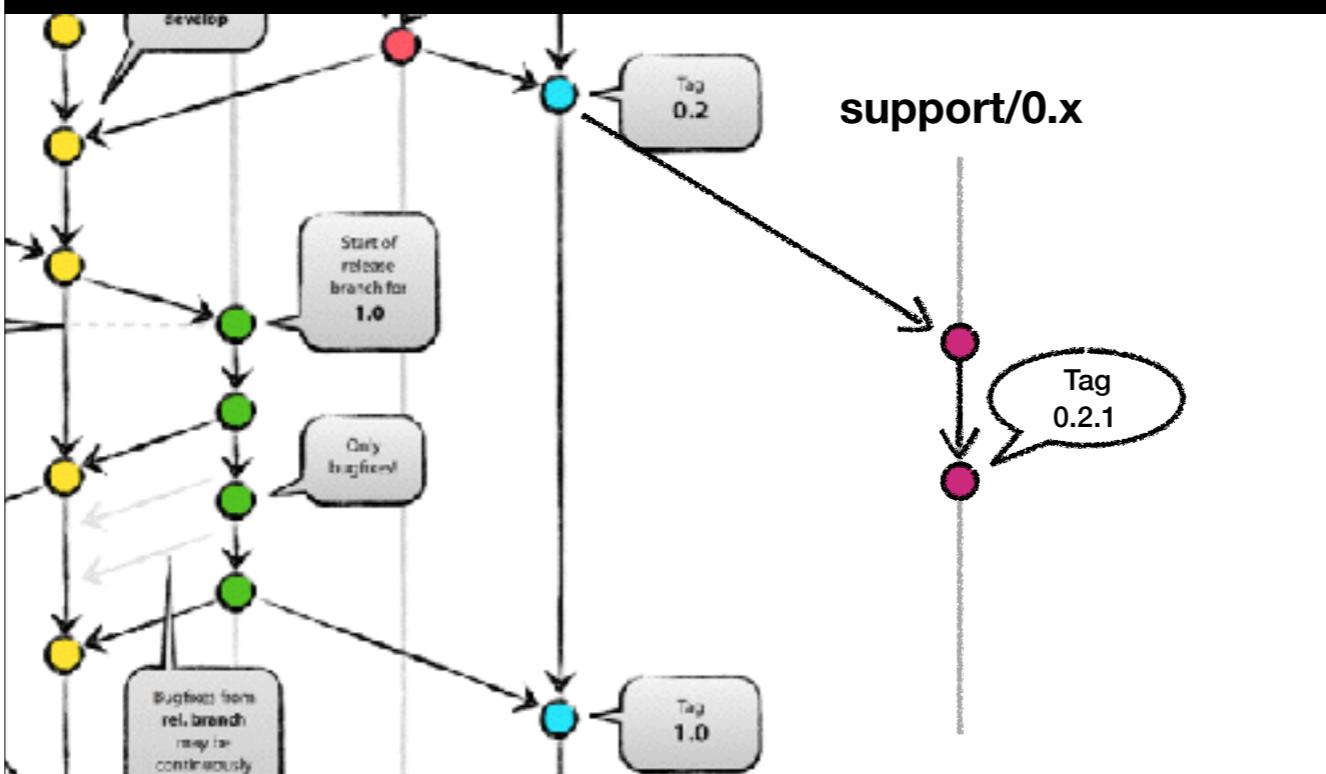


<http://nvie.com/posts/a-successful-git-branching-model/>



<http://nvie.com/posts/a-successful-git-branching-model/>

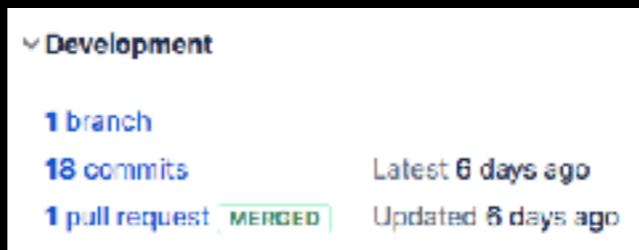
support branches



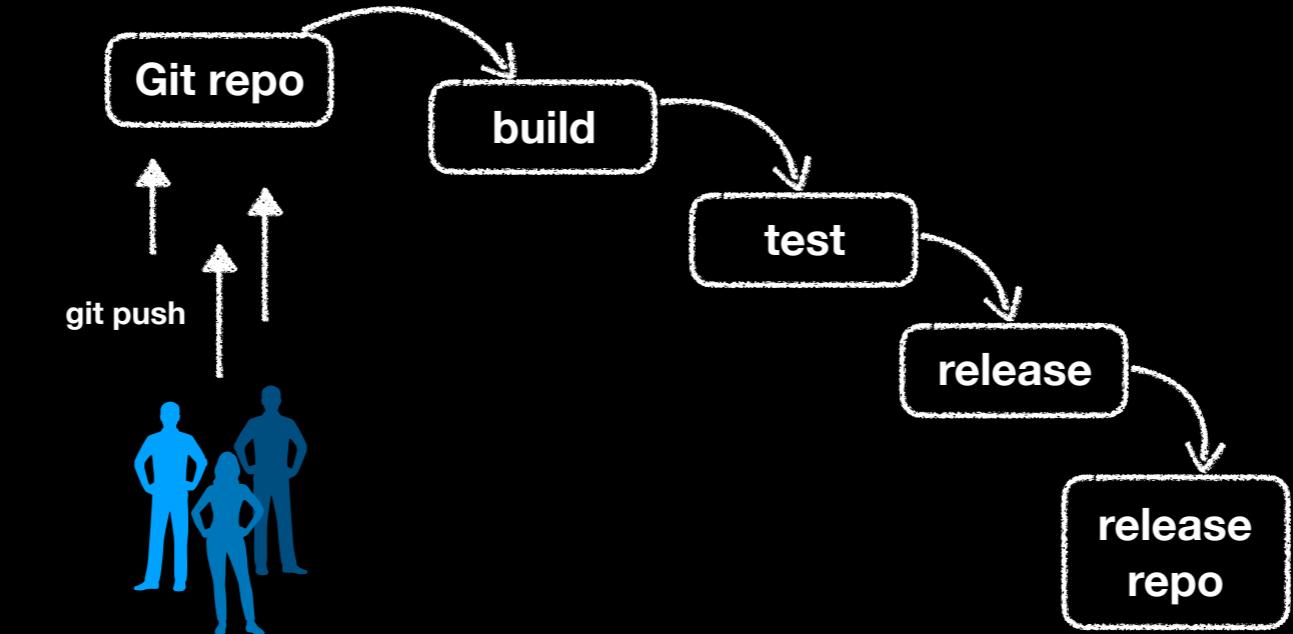
commit <> story

```
git commit -m "CW-1234 fix the world"
```

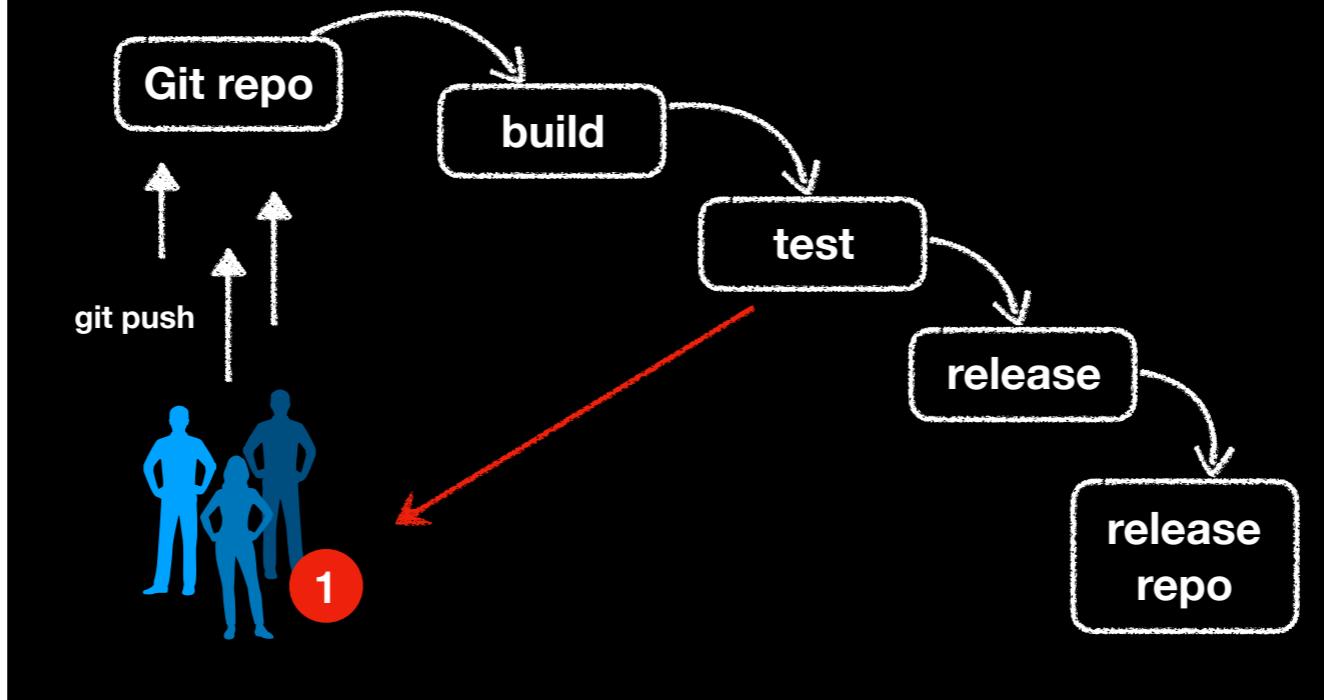
Story CW-1234



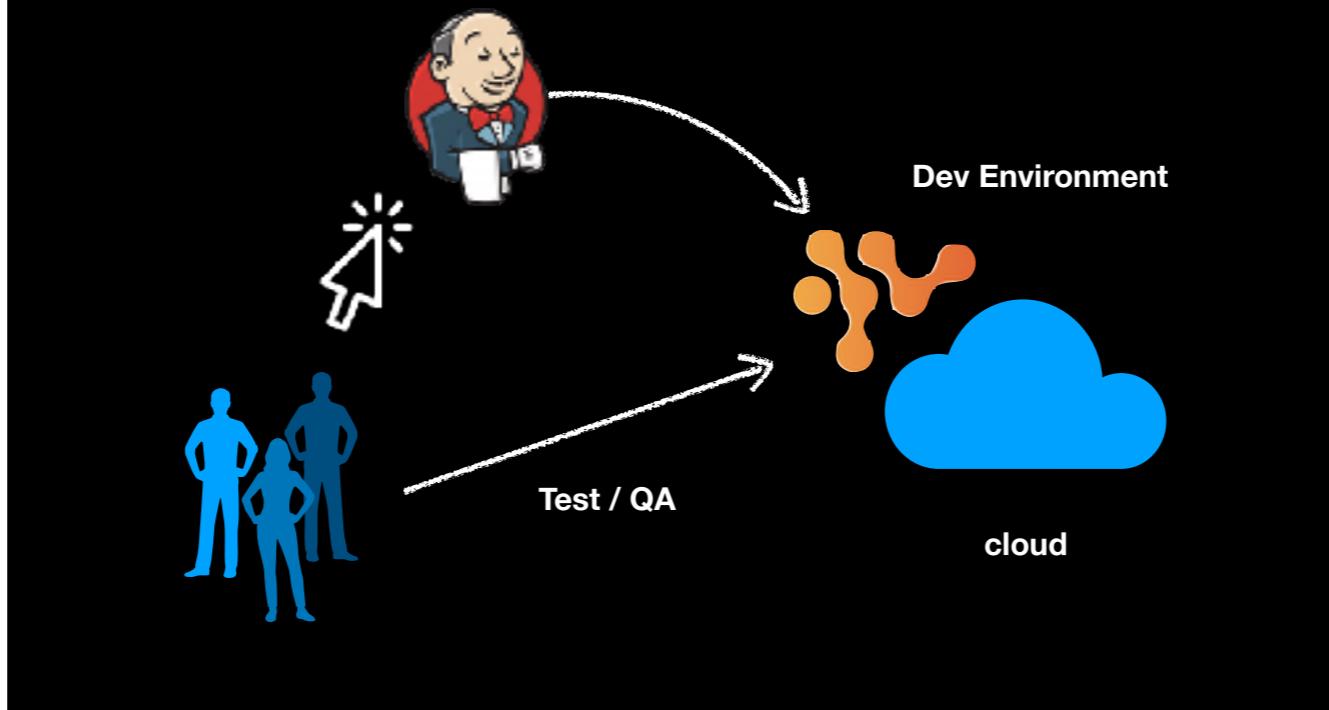
continuous integration



continuous integration



automated deployments



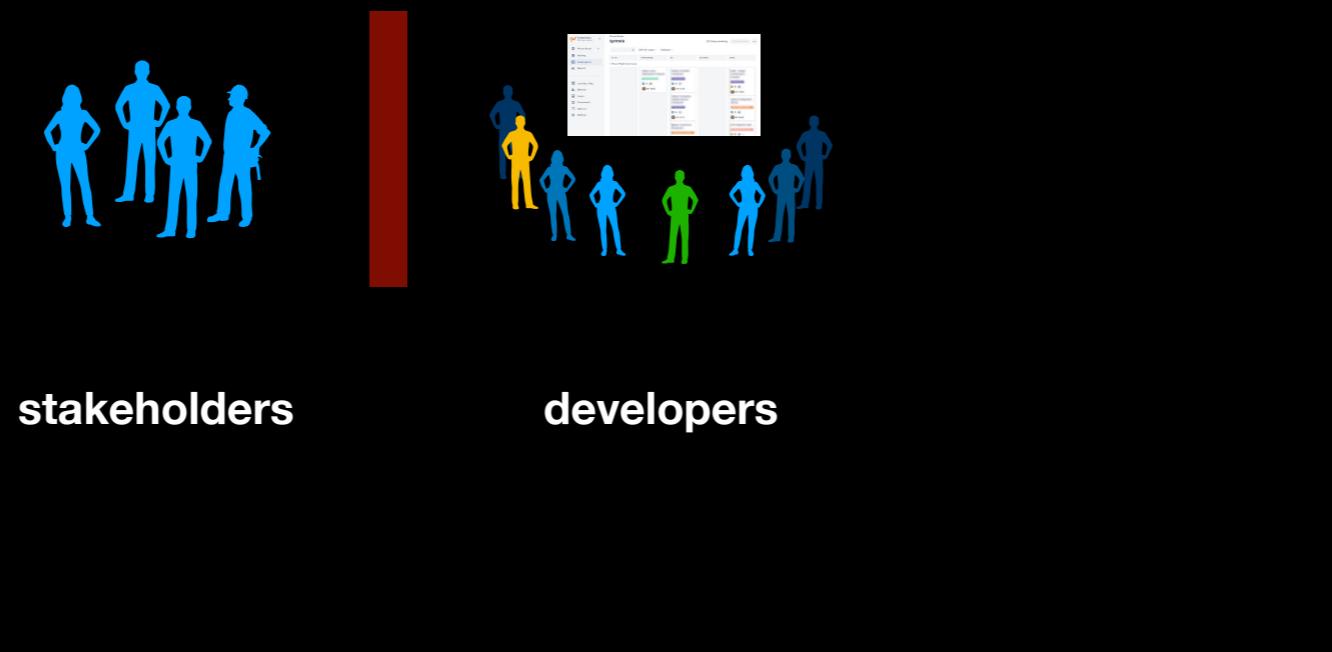
perfect right? what can go wrong... let's deliver!

**WORKED FINE IN
DEV**

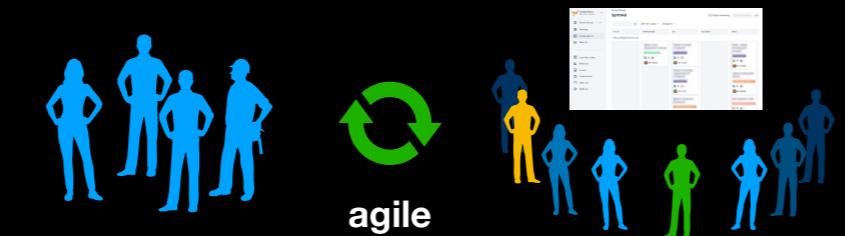
OPS PROBLEM NOW

memegenerator.net

there's another wall to break



there's another wall to break



stakeholders

developers

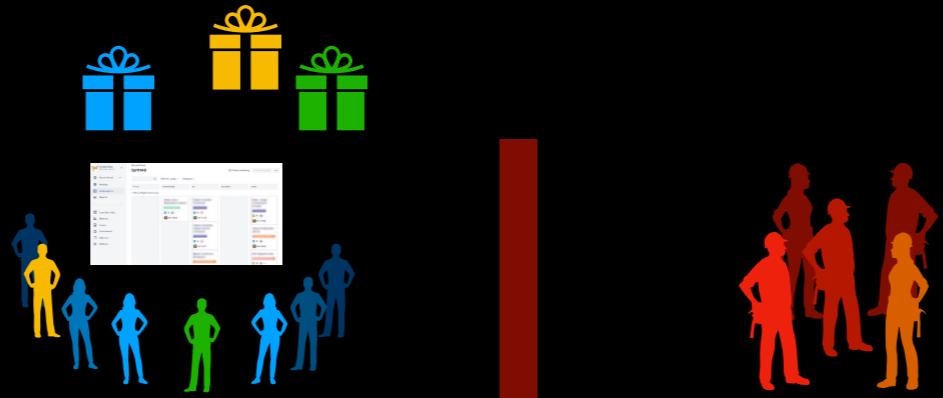
there's another wall to break



there's another wall to break



the wall of confusion



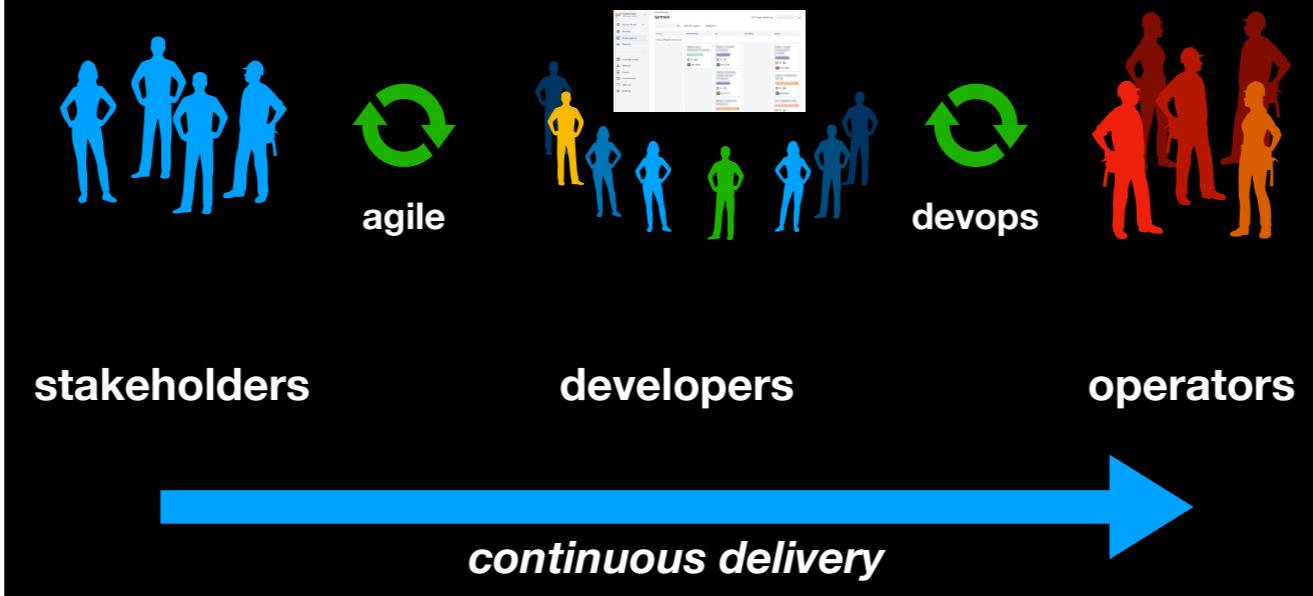
the wall of confusion



devops to the rescue



devops to the rescue



DevOps

- Shared goal: release new features fast and reliability
- How?
 - collaboration
 - common tools
 - automation
 - monitoring
- Long story short: agile with the operations included

From the authors of *The Visible Ops Handbook*



The Phoenix Project

A Novel About IT, DevOps,
and Helping Your Business Win

Gene Kim, Kevin Behr, and George Spafford

If you really want to understand how badly things can go wrong, you should read this novel... a great book about an IT company, written by 3 luminaries of the DevOps movement telling a story which anyone that has been working in IT can recognise... and just read it to know how it worked out.

Infrastructure as code

```
- name: Install Java 8
  become: yes
  package:
    name: openjdk-8-jre
    state: present

- name: Add Cassandra debian repo key
  become: yes
  apt_key:
    url: "{{ cassandra_repo_key }}"
    state: present

- name: Add Cassandra repo
  become: yes
  apt_repository:
    repo: "{{ cassandra_repo }}"
    state: present

- name: Install Cassandra
  become: yes
  apt:
    name: cassandra
    state: present
  register: cassandra_installation

- name: Wait for cassandra to be up
  wait_for:
    ports: 7190
  when: cassandra_installation.changed
```



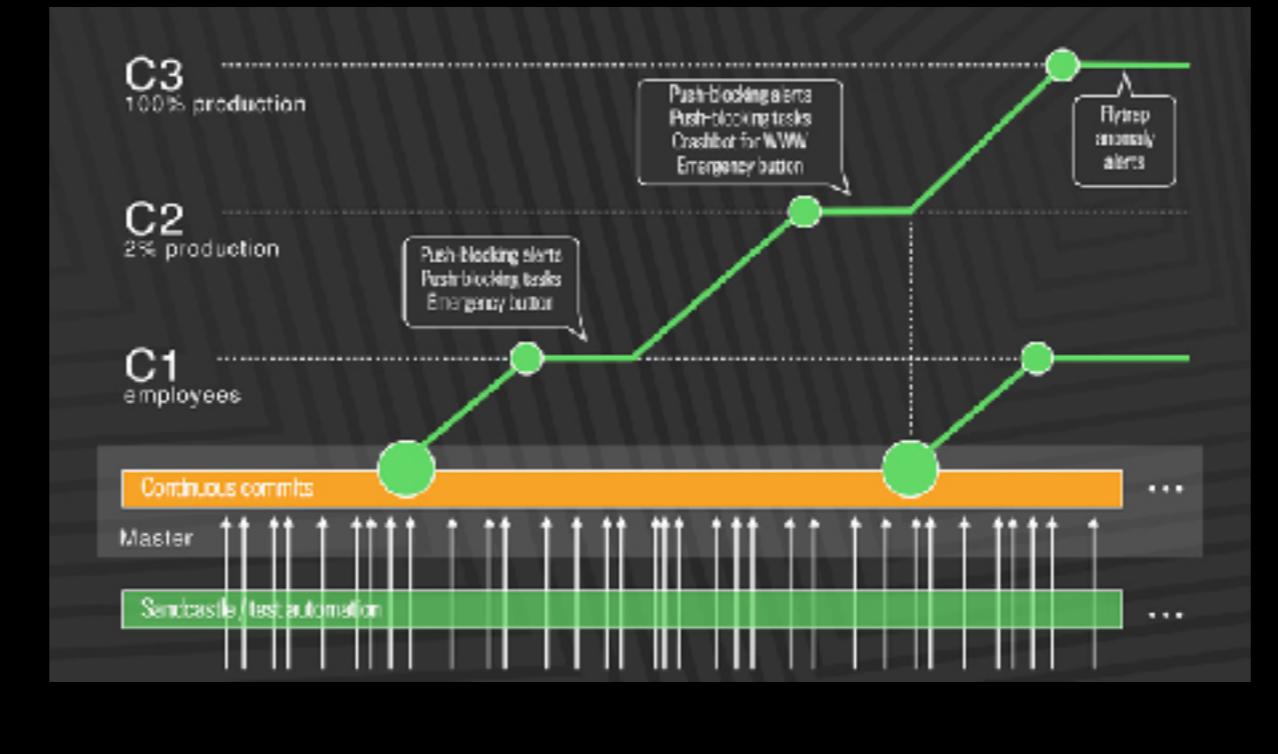
continuous delivery

- software can be released to production at any time
- prioritising keeping the software deployable over working on new features
- automated feedback on the production readiness
- push-button deployments of any version of the software to any environment on demand

Credits: Martin Fowler

Martin Fowler: <https://martinfowler.com/bliki/ContinuousDelivery.html>

continuous delivery at facebook

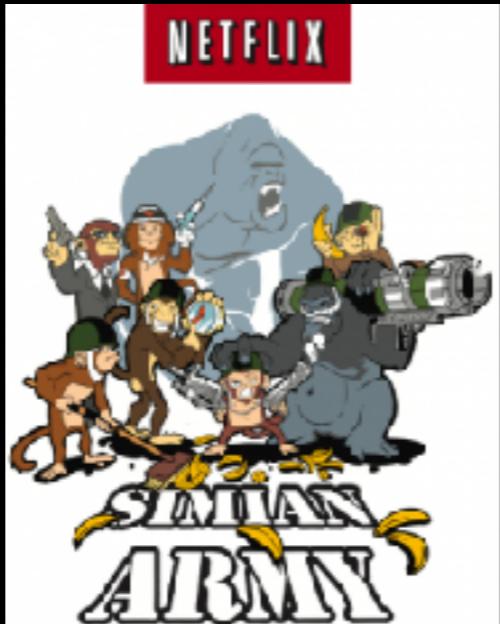


<https://code.facebook.com/posts/270314900139291/rapid-release-at-massive-scale/>

lightning fast

- Etsy: 50 deployments per day
- Netflix: thousands of time per day
- **Amazon: new release every 11.7 seconds**

embracing fragility



Either you break it,
or users will

thank you



I hope I didn't bore you too much. I was asked to share my experience to you and I was happy to accept because, I hope you noticed, I'm really passionate of what I do and I find computer science such an awesome field... and when you own something beautiful you can't keep it for yourself, you love to share it and talk about it to everyone... even to your kids.

Also, my passion for computer science didn't originate from nothing, it was fed by meeting and working with passionate people. That's why I'm here now, hopefully to at least some of you I told something useful. Maybe not, I felt like I had to try though.

Also I shared some of the challenges, some of my experience as a student, something for sure many of you perfectly understand and seems normal... however is all part of who I am now, and maybe some of you is living the same experience and listening that someone else passed from there and survived is encouraging or may originate questions I will be happy to answer now or, better, in front of a beer.

suggested follow-ups

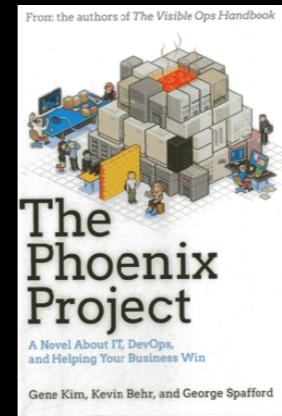
Agile / DevOps luminaries

@martinfowler @RealGeneKim @mitchellh
@patrickdebois @adrianco @botchagalupe

technologies



book



feel free to contact me:
marco.miglierina@icloud.com