

Continuous Architecting Demystified!

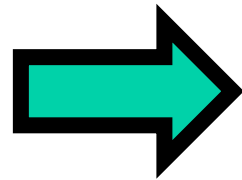
Vienna Software Seminar '17

Damian A. Tamburri
Technical University Eindhoven and
Jeronimus Academy of Data Science (NL)

My Wheel of Life



B.Sc.
**Formal Languages
and Methods for
Software Analysis,
Design and Testing**

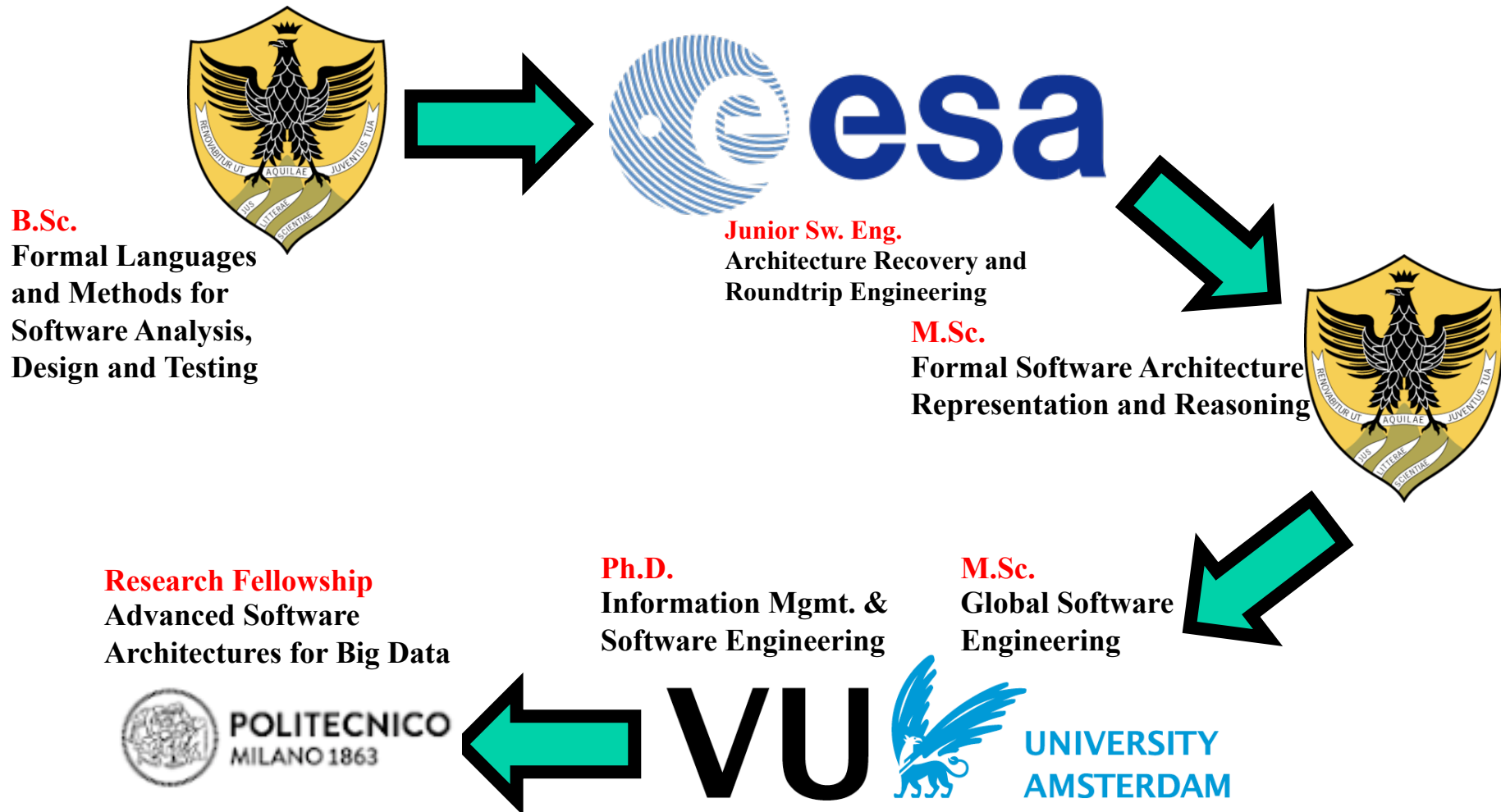


Junior Sw. Eng.
**Architecture Recovery and
Roundtrip Engineering**

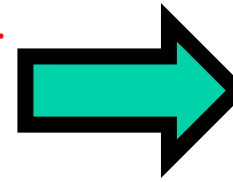
Mission Accomplished*!



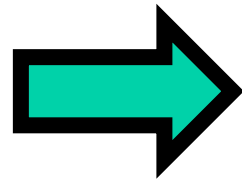
That was a lot of hair ago...



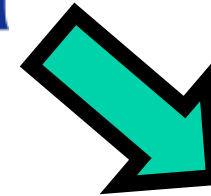
Currently: Assistant Professor
Socio-Technical
Intelligence



B.Sc.
Formal Languages
and Methods for
Software Analysis,
Design and Testing



Junior Sw. Eng.
Architecture Recovery and
Roundtrip Engineering



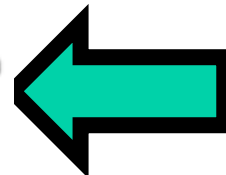
M.Sc.
Formal Software Architecture
Representation and Reasoning



Research Fellowship
Advanced Software
Architectures for Big Data



Ph.D.
Information Mgmt. &
Software Engineering

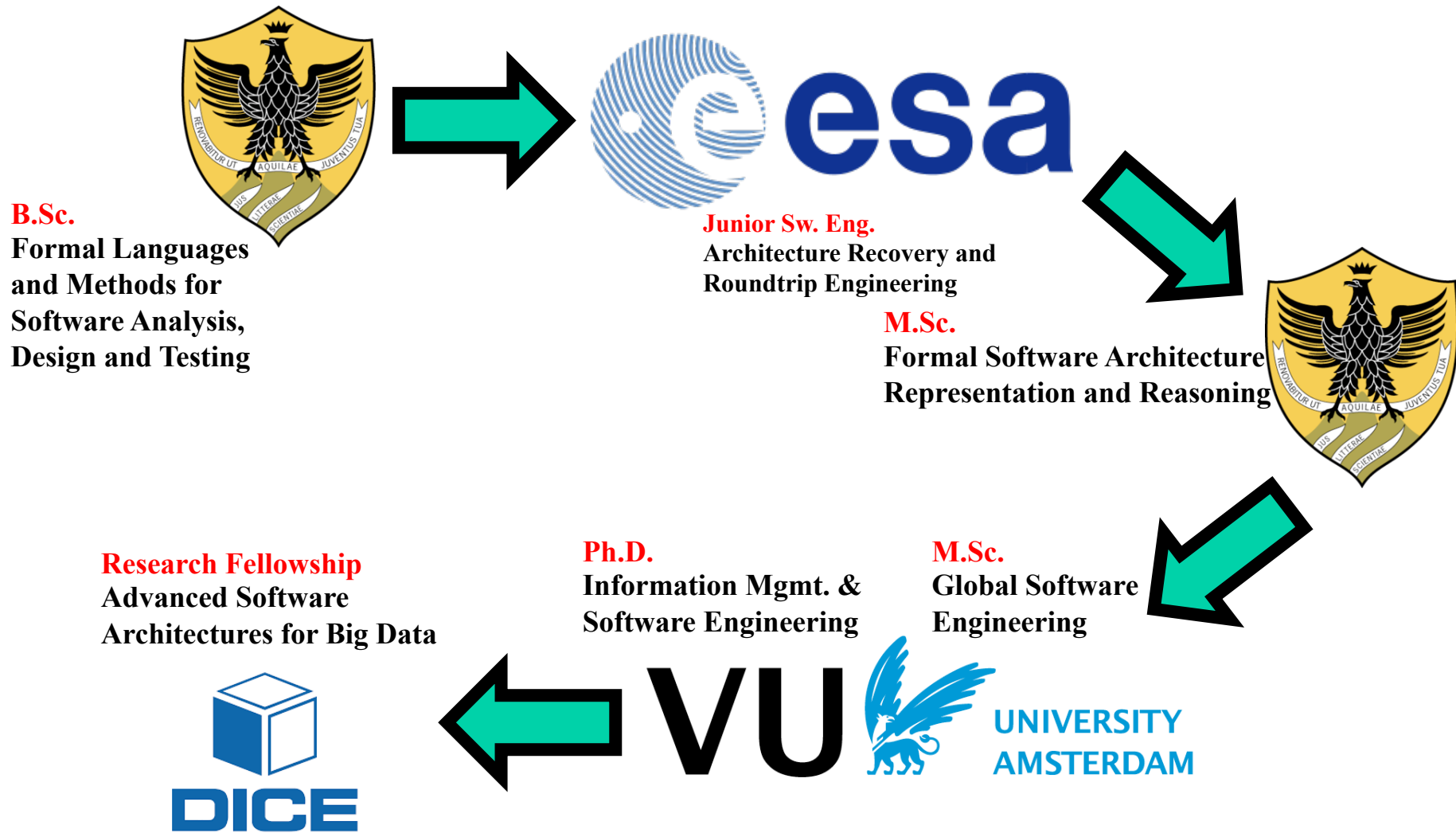


VU



M.Sc.
Global Software
Engineering

**UNIVERSITY
AMSTERDAM**



Hot Topic for Today!

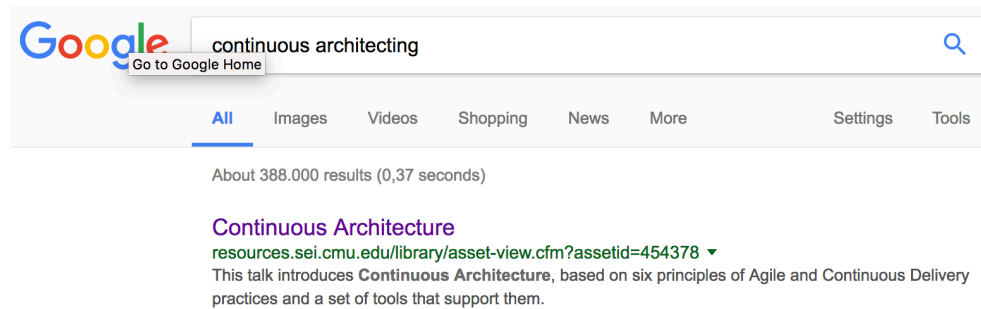
-
- Continuous Architecting!

Hot Topic for Today!

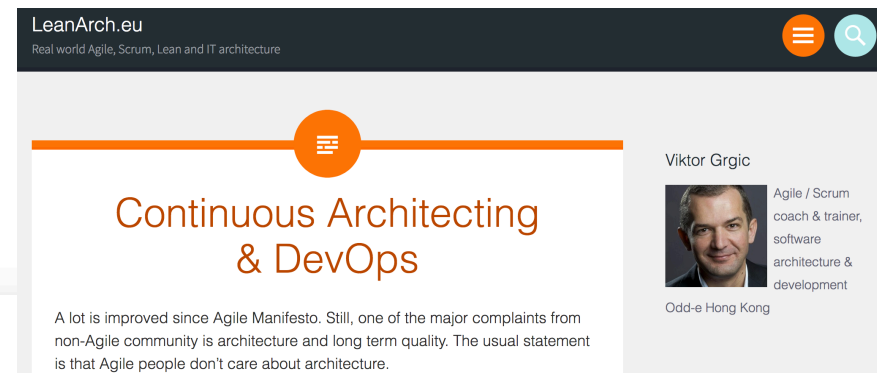
- Continuous Architecting!
 - ▶ “Say What???????”



Not something I, or EU DICE invented...



Continuous architecture in a large distributed agile organization - a case study at Ericsson



Let's check the status of your face...

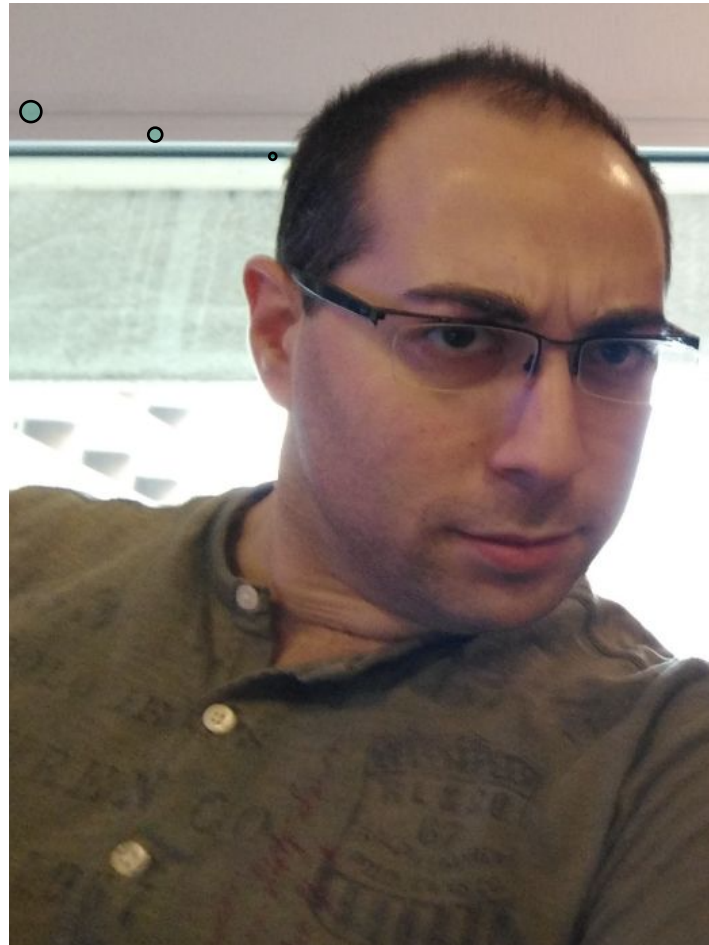
- Continuous Architecting!
 - ▶ “Say What???????”



My Face when I heard of it...

- Continuous Architecting!

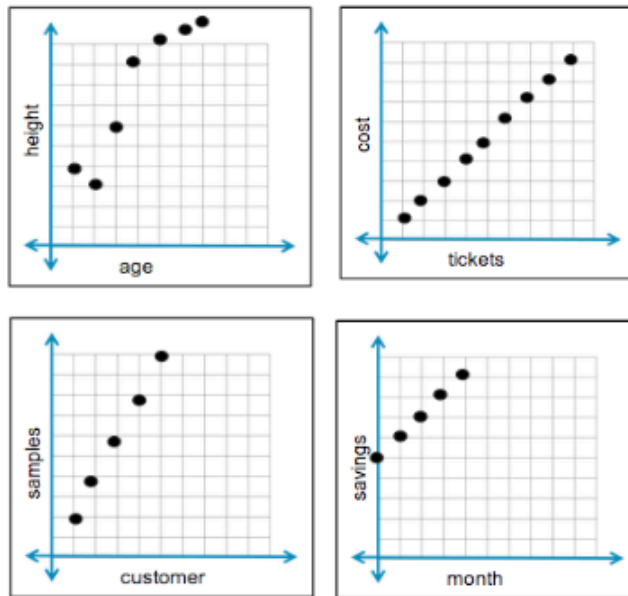
▶ “Say What???????”



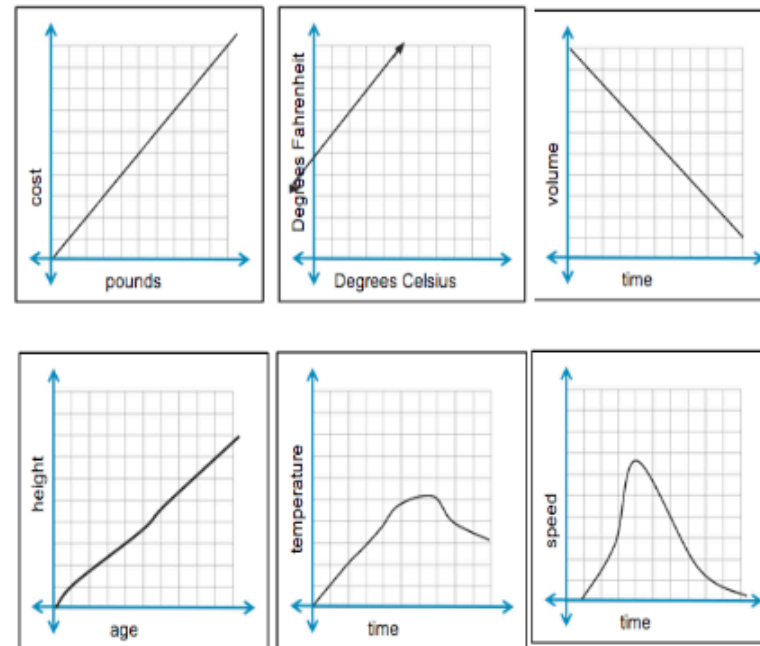
Hot Topic for Today!

- Continuous Architecting!

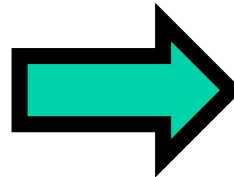
DISCRETE



CONTINUOUS



SAs Before...

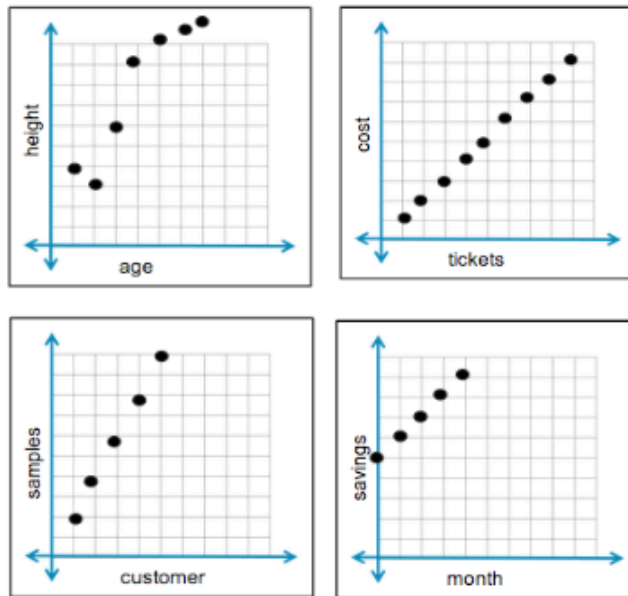


SAs After!

Hot Topic for Today!

- **Software Architectures**

DISCRETE



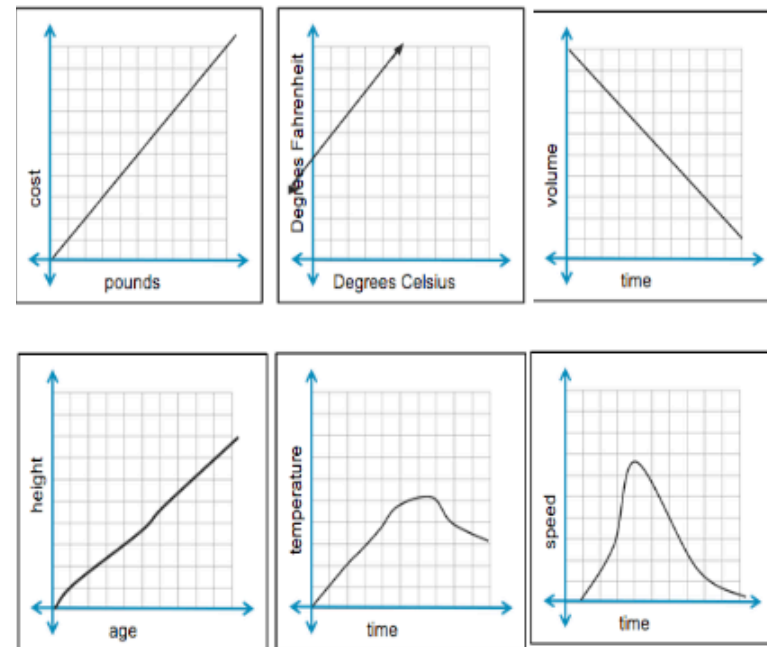
- Set of design decisions;
- Assessed before starting implementation, then changed during lifecycle;
- Documented;
- ...

SAs Before...

Hot Topic for Today!

- **Continuous Architecting!**
 - ▶ Architecture Decisions are not taken, *they *emerge* in a *data-driven* fashion*;
 - ▶ Decision-Making is “Just-in-time”, only where and when **extremely needed**;
 - ▶ Make everything as a **product**, leveraging the **small** (Microservices);
 - ▶ ...

CONTINUOUS



SAs Before...

Hot Topic for Today!

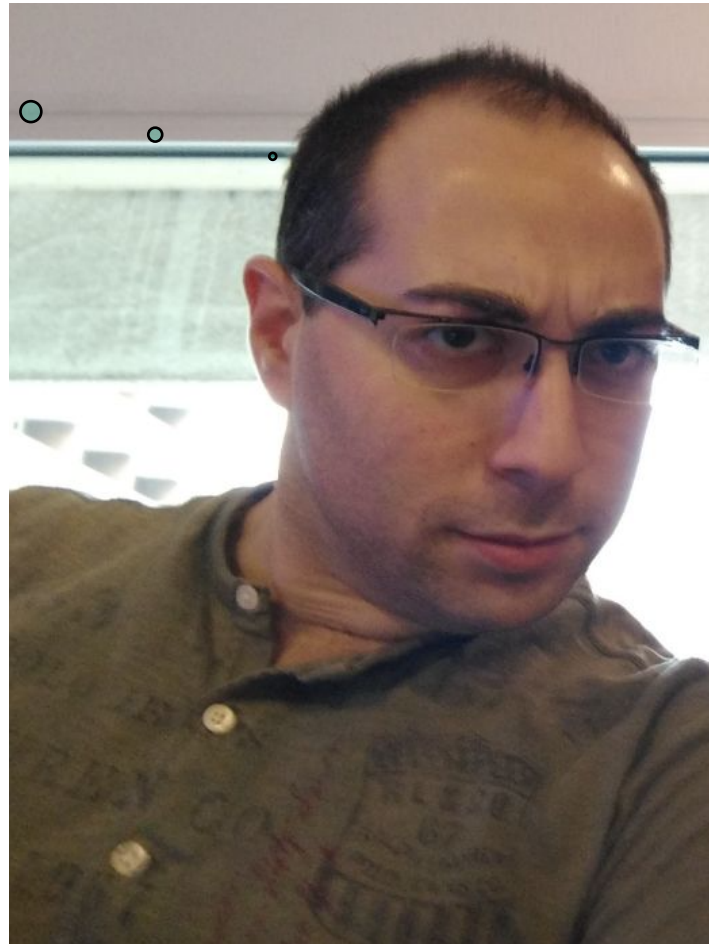
- ▶ “Say What???????”



My Face when I heard of it...

- Continuous Architecting!

▶ “Say What???????”



But first... A bit of history!

- Let's take a step back to where it all began...



It's 2013...

-
- And...

It's 2013...

- And...



It's 2013...

-
- And... EU election 2014: Italy's Renzi triumphs as comic Grillo loses ground

New PM scores sweeping victory in election, leaving former comic Beppe Grillo's anti-establishment 5-Star Movement and Silvio Berlusconi's Forza Italia trailing



Meanwhile in Software Engineering...

Top failure causes*

- Unrealistic deadlines, e.g., imposed by someone external to the technical staff
 - Requirements & people change (too) often
 - Effort and resources have been estimated in an overly optimistic way,
 - Risks have not been taken into account from the start of the project.
 - Risks can be technical or human difficulties
 - Communication problems among staff members
 - Difficulty by the management to recognize recurrent delays and take immediate action
 - Subversive stakeholders
-

Meanwhile in Software Engineering...

Top failure causes* - **An Example!**

- **Unrealistic deadlines, e.g., imposed by someone external to the technical staff**
 - **Requirements & people change (too) often**
 - Effort and resources have been estimated in an overly optimistic way,
 - Risks have not been taken into account from the start of the project.
 - Risks can be technical or human difficulties
 - **Communication problems among staff members**
 - Difficulty by the management to recognize recurrent delays and take immediate action
 - **Subversive stakeholders**
-

Meanwhile in Software Engineering...

Top failure causes* - **An Example!**

(UNFORESEEN) OVERHEAD COST: 174,000,000 \$ (give or take)*

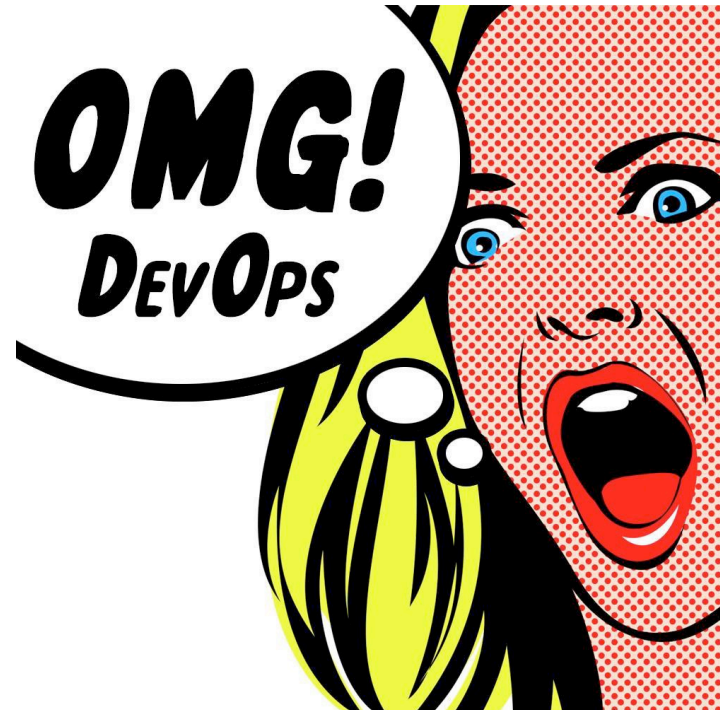


*<http://www.cio.com/article/2380827/developer/6-software-development-lessons-from-healthcare-gov-s-failed-launch.html>

What is DevOps?

“DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.”

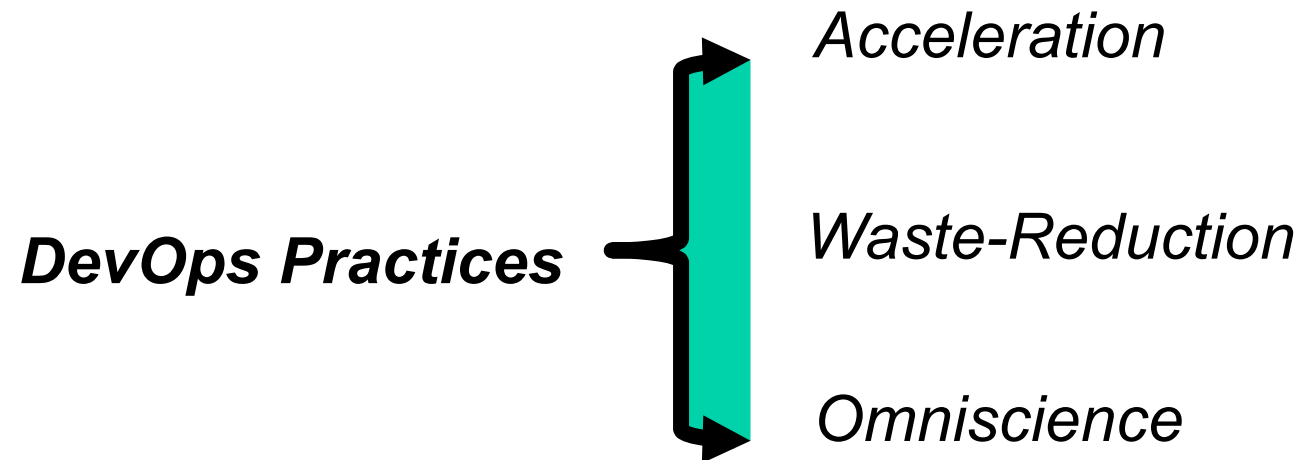
L. Bass et Al. [11]



What is DevOps?

*“DevOps is a set of **practices** intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.”*

L. Bass et Al. [11]



DevOps Practices: Let's take a look



- **Acceleration Tactics**

- ▶ Use Faster Organization: Merge Dev+Ops Teams...
- ▶ Infrastructure-as-Code
- ▶ Use Continuous Integration Tools
- ▶ Use Continuous Deployment Tools
- ▶ Use Continuous Testing Tools
- ▶ ...

- **Waste Reduction Tactics**

- ▶ Canary Testing
- ▶ A/B Testing
- ▶ Reduce Documentation
- ▶ Minimalistic Architecting → Microservices
- ▶ ...

DevOps Practices: Let's take a look



- **Omniscience Tactics**
 - ▶ Monitor Everything
 - ▶ Monitoring-as-a-service
 - ▶ On-The-Fly Risk Engineering
- ...

DevOps Practices: Let's take a look



- **Omniscience Tactics**
 - ▶ Monitor Everything
 - ▶ Monitoring-as-a-service
 - ▶ On-The-Fly Risk Engineering
 - ▶ **Continuous Architecting!**
- ...

- Software Architecture responds to architecture drivers... So... “Just” upgrade the drivers for DevOps!
 - ▶ Design for Modifiability
 - ▶ Design for Observability
 - ▶ Design for Organisability
 - ▶ Design for Fast Evolution & Testability
 - ▶ Design for High Scalability

but... most of all...

- Software Architecture responds to architecture drivers... So... “Just” upgrade the drivers for DevOps!

- ▶ Design for Modifiability
- ▶ Design for Observability
- ▶ Design for Organisability
- ▶ Design for Fast Evolution & Testability
- ▶ Design for High Scalability

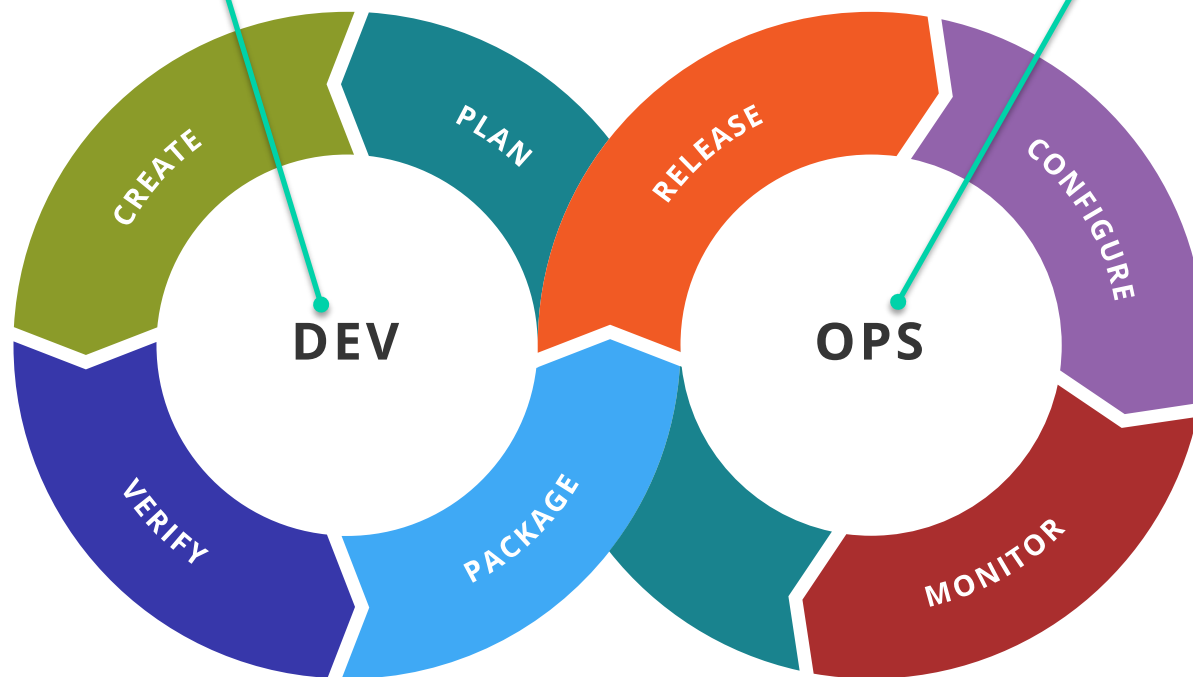
but... most of all...

- ▶ **Design for SA failure!**
 - SA is incremental, refined from a rough draft via **neverending continuous architectural improvement!**

Continuous Architecting In Context

Dev Goal: “Prepare a Software Architecture designed to be immediately deployable”

Ops Goal: “Observe the architecture runtime and provide Ops feedback to Dev... *then improve architecture continuously*”



DevOps processes and toolchain: Putting it all together...

- Continuous Architecting

Def. “architect for test, build and deploy, take quality attributes into account, take advantage of feedback from runtime” [1]

- Continuous Integration

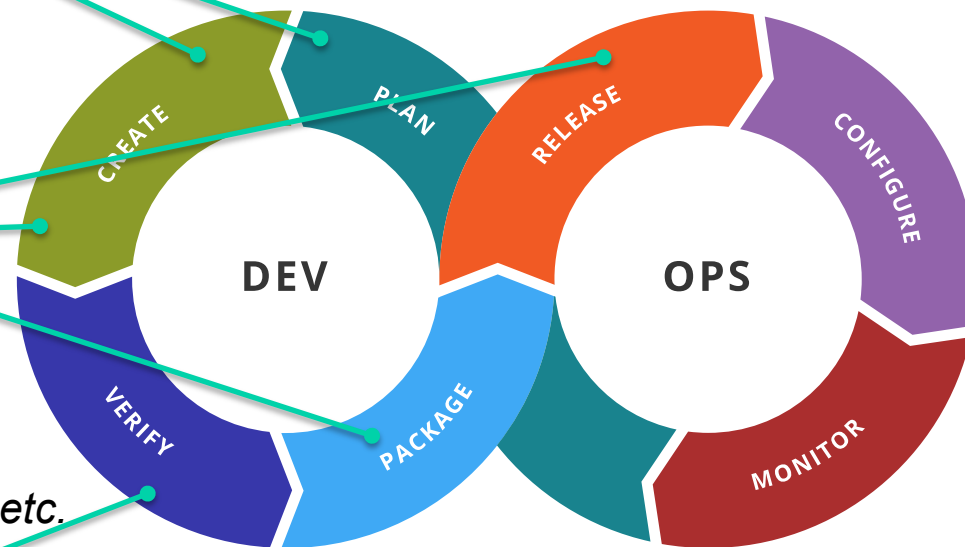
Def. “merge all developer work-copies to a shared mainline frequently” [4]

Examples. *Apache Jenkins, Hudson, etc.*

- Continuous Testing

Def. “run tests as part the build pipeline so that every check-in and deployment is validated” [3]

Examples. *Selenium+GitHub+LI-API, etc.*



...

- Bass et Al. '15 [11]:
 - ▶ **Ops Shift-Left** – “design the SA to cater for continuous deployment, using Infrastructure Design Patterns, IaC, ...”
- Erder & Murat '16 [1]
 - ▶ **Architecture-Driven Organisation** – “Design Architecture for DevOps *then* organise to match it”
 - ▶ **Just-In-Time SA Decision-Making** – “Delay design decisions until they are absolutely necessary”

Hold on a second... Hasn't anyone said this already!?

- Bass et Al. '15 [11]:
 - ▶ **Ops Shift-Left** – “design the SA to cater for continuous deployment, using Infrastructure Design Patterns, IaC, ...”
- Erder & Murat '16 [1]
 - ▶ **Architecture-Driven Organisation** – “Design Architecture for DevOps *then* organise to match it”
 - ▶ **Just-In-Time SA Decision-Making** – “Delay design decisions until they are absolutely necessary”

- Bass et Al. [11]:
 - ▶ Ops Shift-Left – “design the SA to cater for continuous deployment, using Infrastructure Design Patterns, IaC, ...”
- Erder & Murat [1]
 - ▶ Architecture-Driven Organisation – “Design Architecture for DevOps *then* organise to match it”
 - ▶ Just-In-Time SA Decision-Making – “Delay design decisions until they are absolutely necessary”
- Van Vliet et al. [8] **2004!**
 - ▶ **Architecture-Level Modifiability Analysis (ALMA)** – “measure modifiability, (re-)architect to improve it”

...

What changes?

-
- Now it's cool!

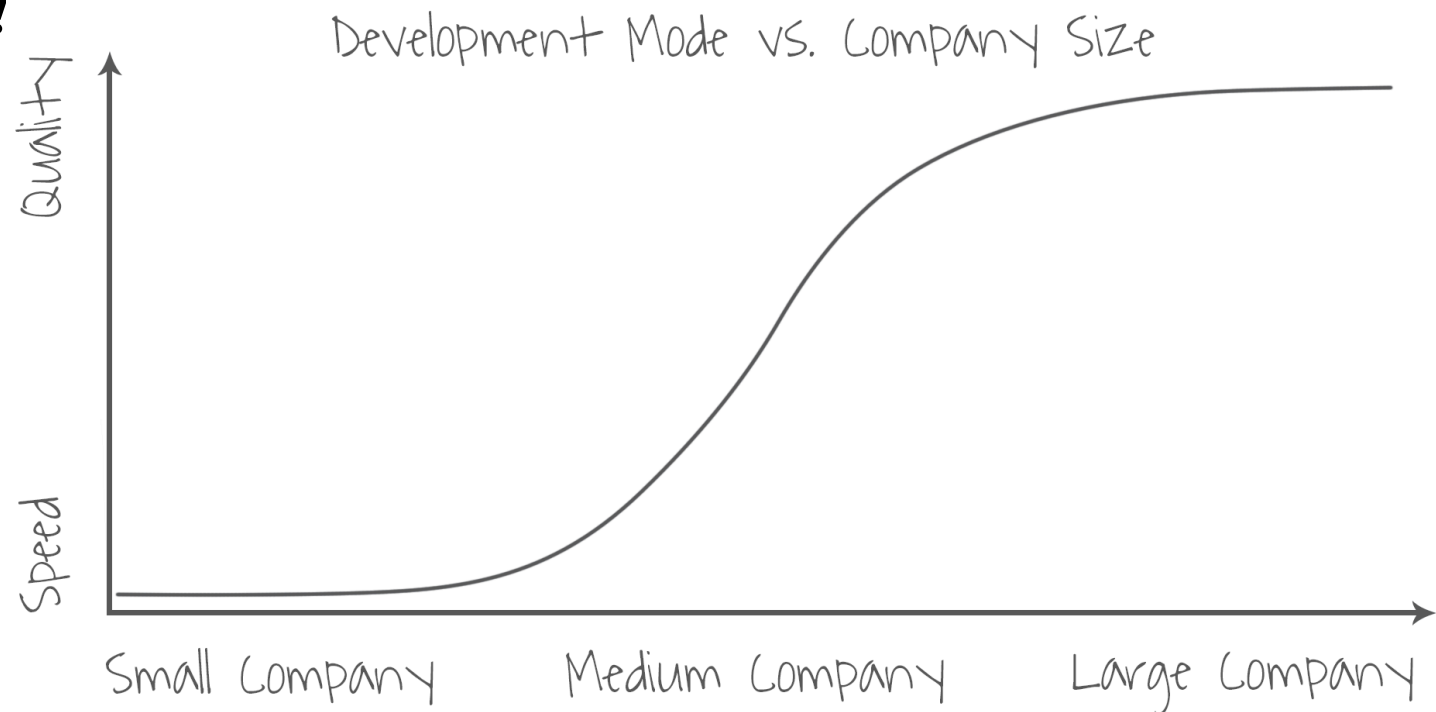
And most importantly...

What changes?

- Now it's cool!

And most importantly...

- **Speed!**



- New architecture drivers:
 - ▶ Modifiability
 - ▶ Observability
 - ▶ Organisability
 - ▶ Speed
 - ▶ Failure
 - ▶ ...

- “New” architecture drivers:
 - ▶ Modifiability
 - ▶ Organisability
- Actually New:
 - ▶ **Observability**
 - ▶ **Speed**
 - ▶ **Failure**
 - ▶ ...

-
- New architecture drivers:
 - ▶ Modifiability
 - ▶ Observability
 - ▶ Organisability
 - ▶ Speed
 - ▶ Failure
 - ▶ ...
 - Continuous Architecting – “more of the same, ***only faster***”

- What we miss, architecturally:
 - ▶ A better connection between the design of software, the design of the infrastructure, and the design of the organization;
 - ▶ A precise and rigorous comparison approach between the new languages and tools for coding infrastructures;
 - ▶ Metrics to track and evaluate all of the above;

- Who is the Architect?
 - ▶ Anyone who enables Continuous-*
 - ▶ Anyone who enables an Agile Organisation
 - ▶ Anyone who enables for new Arch. Drivers' equivalent metrics (e.g., Observability, Modifiability)
 - ▶ So... The architect is a Community Shepherd → but it can be anyone

“The Architect is my Shepherd [...]”



That's all folks!

Any Questions?



References

-
- [1] Erder, Murat and Pureur, Pierre. Continuous Architecture: Sustainable Architecture in an Agile and Cloud-Centric World. Amsterdam: Morgan Kaufmann, 2016.
 - [2] Continuous Testing Paperback - January 2, 2014 by W. Ariola, C. Dunlop
 - [3] Part of the Pipeline: Why Continuous Testing Is Essential, by Adam Auerbach, TechWell Insights August 2015
 - [4] M. Fowler Continuous Integration,
<https://www.thoughtworks.com/continuous-integration>
 - [5] Chen, Lianping (2015) "[Continuous Delivery: Huge Benefits, but Challenges Too](#)" *IEEE Software*. 32 (2): 50.
 - [6] <http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/csd03/TOSCA-Simple-Profile-YAML-v1.0-csd03.html>
 - [7] P. Lipton, D. Palma, M. Rutkowski, and D. A. Tamburri, "Tosca solves big problems in the cloud and beyond!" *IEEE Cloud*, vol. 21, no. 11, pp. 31-39, 2016.
 - [8] Bengtsson, PerOlof, Lassing, Nico, Bosch, Jan and van Vliet, Hans. "Architecture-level modifiability analysis (ALMA)." *Journal of Systems and Software* 69 , no. 1--2 (2004): 129--147.
 - [9] Tamburri, D. A.; Lago, P. & van Vliet, H. (2013), 'Uncovering Latent Social Communities in Software Development.', *IEEE Software* 30 (1) , 29-36 .
 - [10] M. Di Penta, D. A. Tamburri, Combining Quantitative and Qualitative Methods in Empirical Software Engineering Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM Sigsoft Symposium of the Foundations of Software
-

-
- [11] Bass, L. J.; Weber, I. M. & Zhu, L. (2015), *DevOps - A Software Architect's Perspective.* , Addison-Wesley .
- [12] Tamburri, D. A. & Nitto, E. D. (2015), When Software Architecture Leads to Social Debt., *in* Len Bass; Patricia Lago & Philippe Kruchten, ed., 'WICSA' , IEEE Computer Society, pp. 61-64 .
- [13] Tamburri, D. A.; Kruchten, P.; Lago, P. & van Vliet, H. (2015), 'Social debt in software engineering: insights from industry.', *J. Internet Services and Applications* **6** (1) , 10:1-10:17 .
- [14] Tamburri, D. A.; Lago, P. & van Vliet, H. (2013), 'Organizational social structures for software engineering.', *ACM Comput. Surv.* **46** (1) , 3 .