

Antonio Martini

# MANAGING ARCHITECTURE TECHNICAL DEBT IN AGILE ARCHITECTING



G.U.



CHALMERS

# Who is Antonio Martini?

- ◉ Italian
  - No kebab pizza! ☺
  - 4 years in Sweden – survived many winters!
- ◉ Bachelor in Computer Science
- ◉ Master in Software Engineering
- ◉ Previous work
  - Back-end development
  - GUI development
  - Contact with the customer (“PO”)
- ◉ PhD Licentiate in 2013
- ◉ Now PhD Candidate in Software Engineering
  - Planning to finish soon...
- ◉ Hobbies (and no time to spend on them ☺)
  - Board games, strategy computer games, pool, etc.
  - Football, volleyball, beach volley, fencing
  - Piano, Drumset, etc.
  - Travel!



G.U.



CHALMERS

# A Software Center Project

Current participants from industry



\*\*



\* Several sites at  
Ericsson

\* The project started at Ericsson, especially with the help of the teams from DURA and EPG.

\*\* Also Volvo Powertrain contributed significantly to the first results



G.U.



CHALMERS

# The shift from Waterfall to Agile and the role of Architecture

# Background

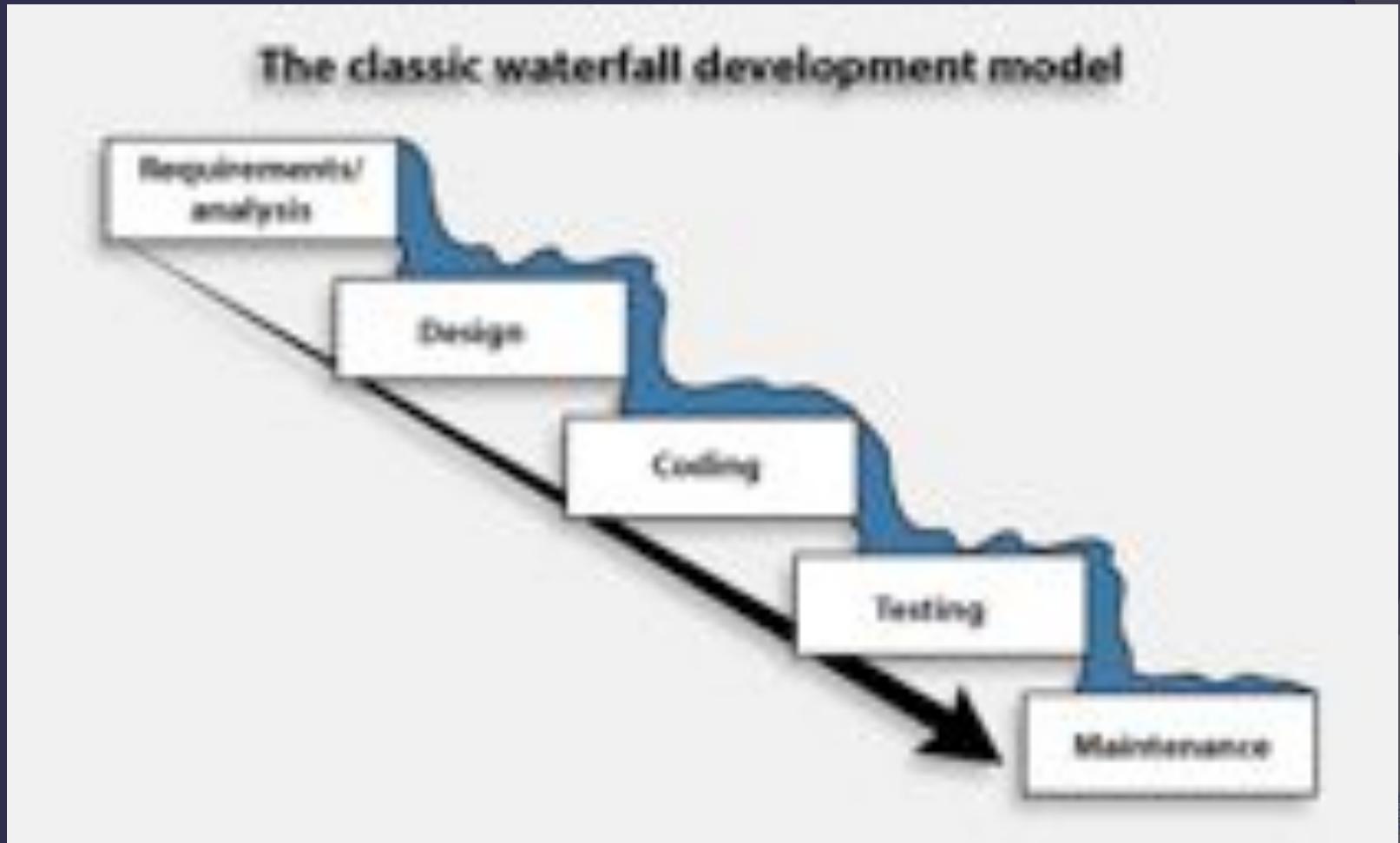


G.U.

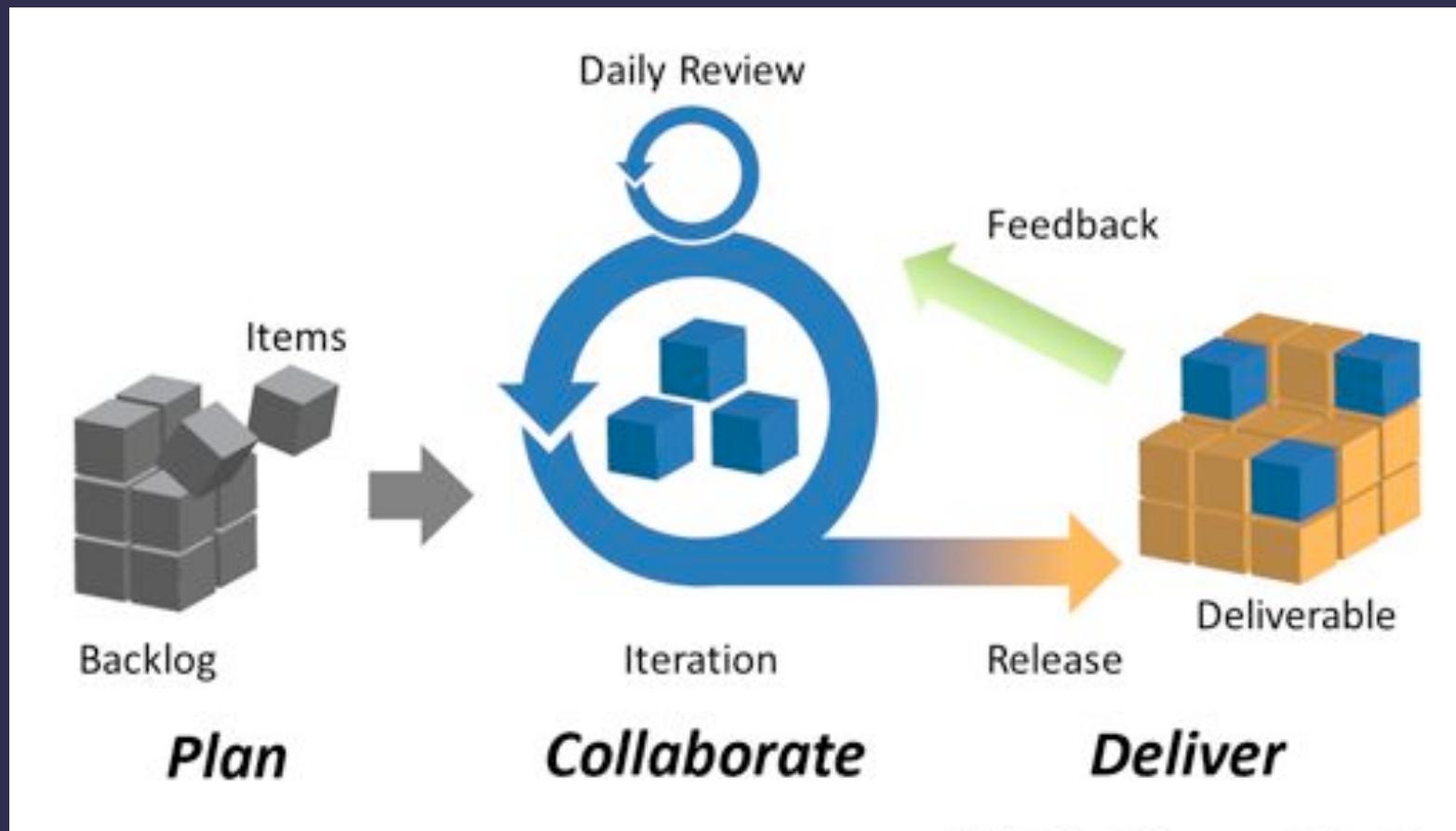


CHALMERS

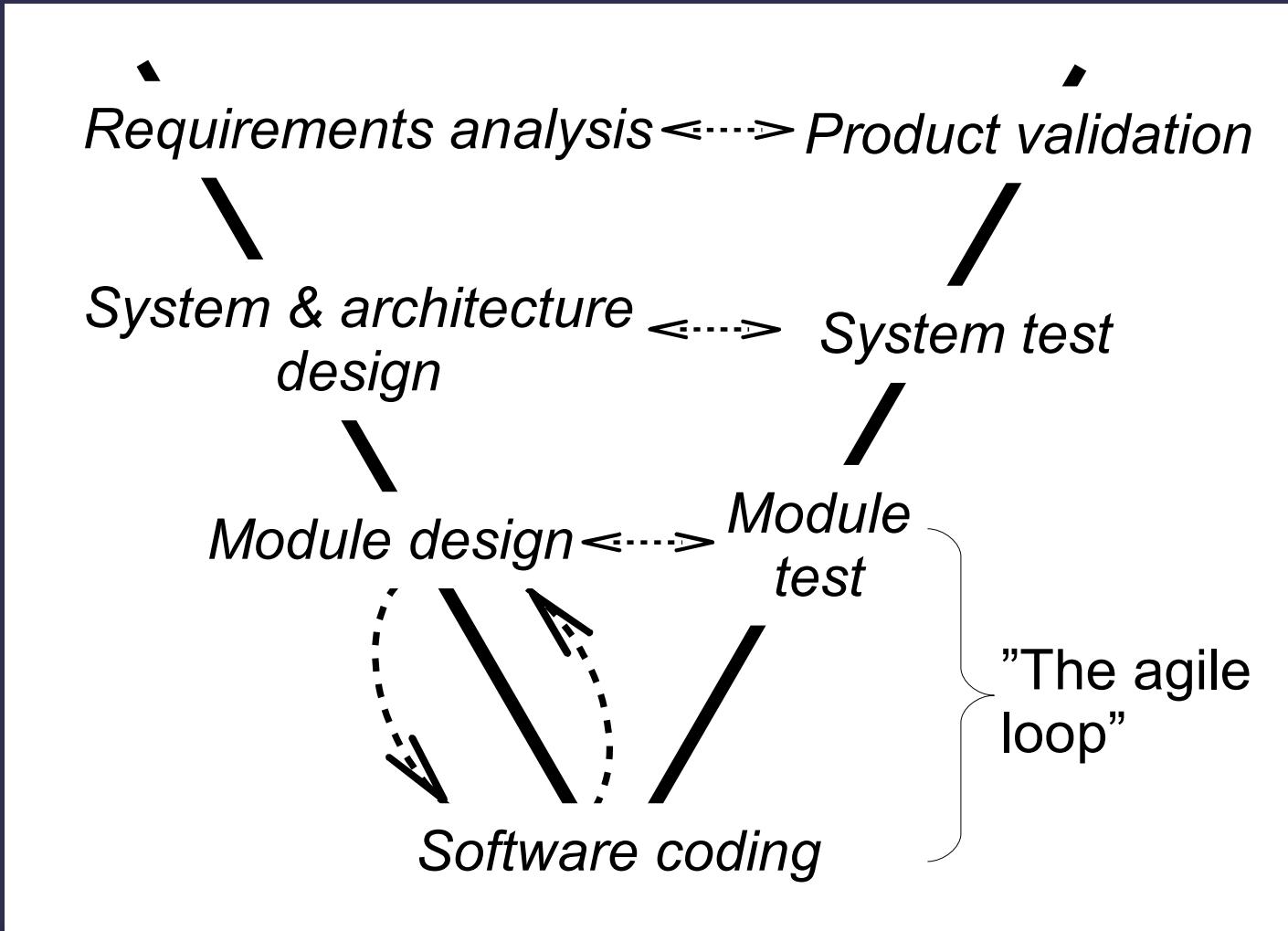
# Once upon a time it was Waterfall...



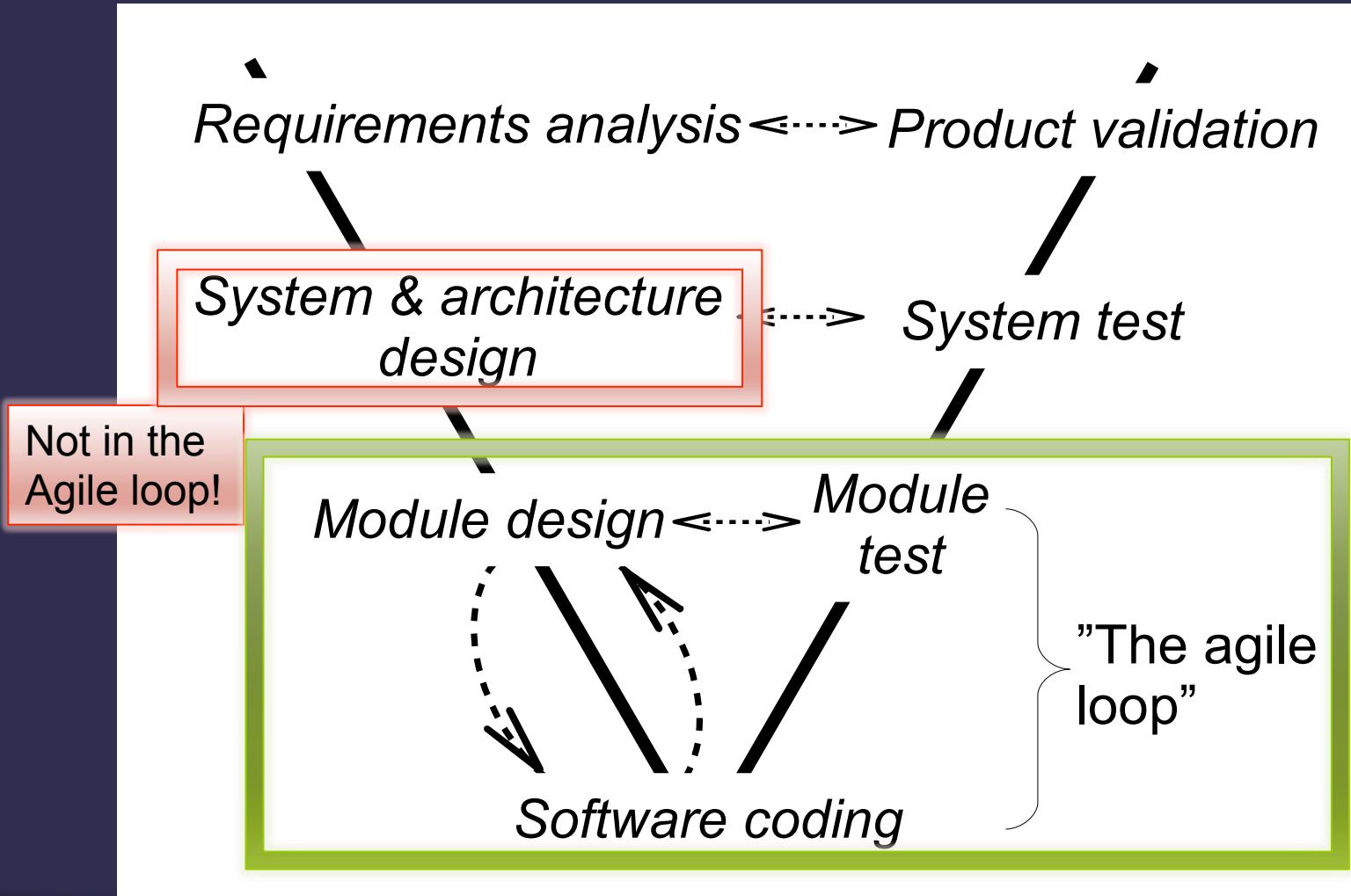
# ...then the Agile revolution happened



# What exactly happened in large companies?



# What about architecture?



# Architecture in Agile?

- ◉ Not recognized in Agile practices
- ◉ “Just enough architecture/design”
- ◉ But what does that mean?



G.U.



CHALMERS

# Agile and Speed

Fast delivery of business value



FT = Feature Team



G.U.



CHALMERS

# Architecture "runway"

Agile teams need to be supported by an ***architecture runway***



"In systems of complexity, building and maintaining sufficient architectural runway is a key to overall productivity"

- Leffingwell\*

## Moral

FTs are great for missions, but someone must provide the runway, for the FTs' takeoffs and returns

\*) Dean Leffingwell, *Scaling Software Agility*, 2007



# Agile and Architecture Runway

## Agile



- Stakeholder orientation
- Responsiveness
- Frequent deliveries
- Light-weight communication

## Architecture Runway



- Structure
- Infrastructure
- Tooling
- Automation
- Education



G.U.



CHALMERS

# Moral from Agile: not too much architecture runway



No delivered value!



G.U.



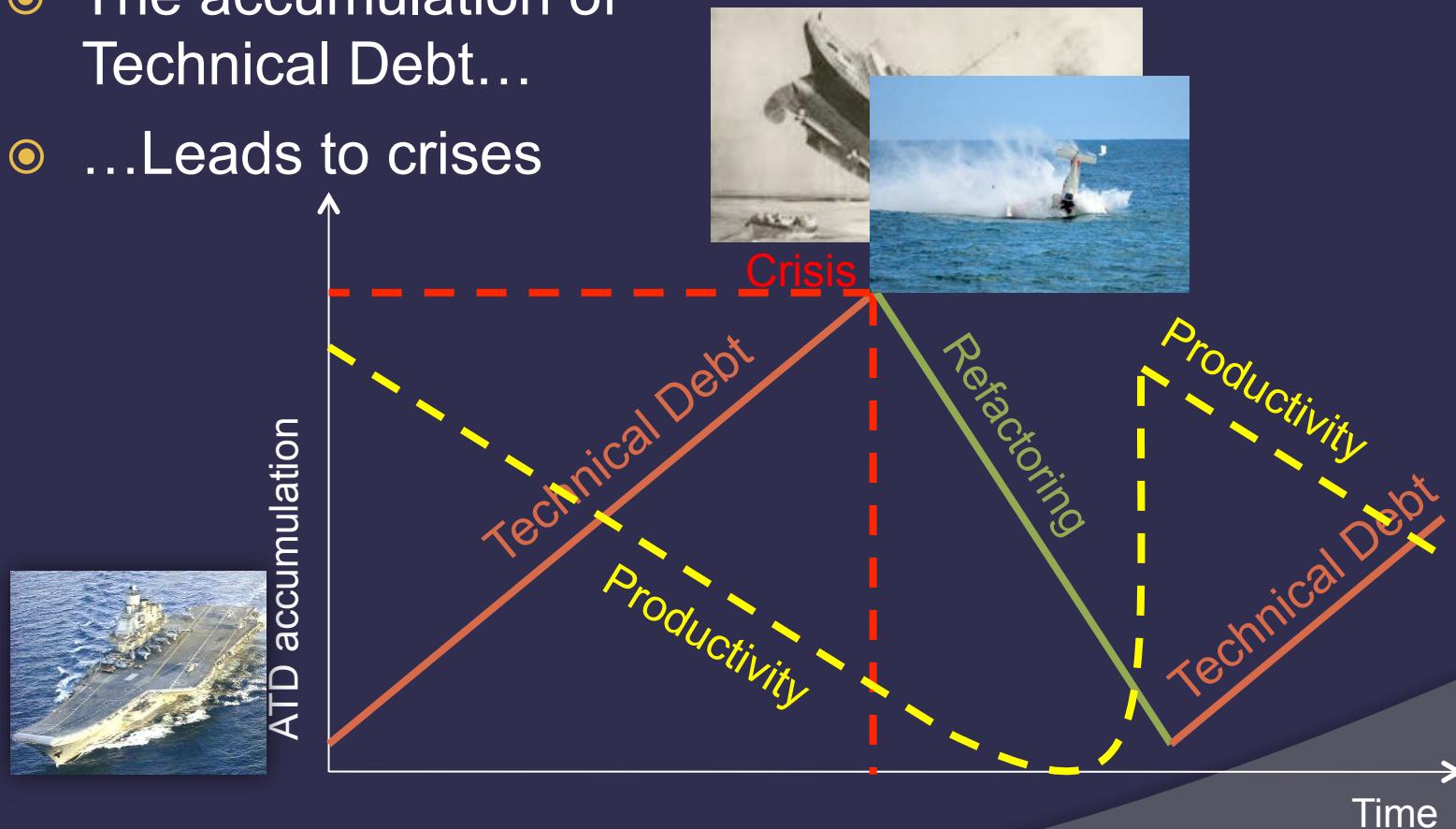
CHALMERS

# But what happens with not enough architecture runway?



# What happens if the runway “decays”?

- Research study in 7 organizations \*
- The accumulation of Technical Debt...
- ...Leads to crises



\* Martini, A., Bosch, J., Chaudron, M., 2014. Architecture Technical Debt: Understanding Causes and a Qualitative Model, in: 40th Euromicro Conference on Software Engineering and Advanced Applications.

# What happens if the runway “decays”?



No delivered value!



G.U.



CHALMERS

The danger hides in everyday decisions

# Architecture Technical Debt



G.U.



CHALMERS

# What is Architectural Technical Debt?

- ◉ Sub-optimal architectural solutions that
  - have a beneficial impact on short-term goals but (taking debt)
  - Have a negative impact in the medium-long run (paying the interest)
- ◉ Better explanation: a story



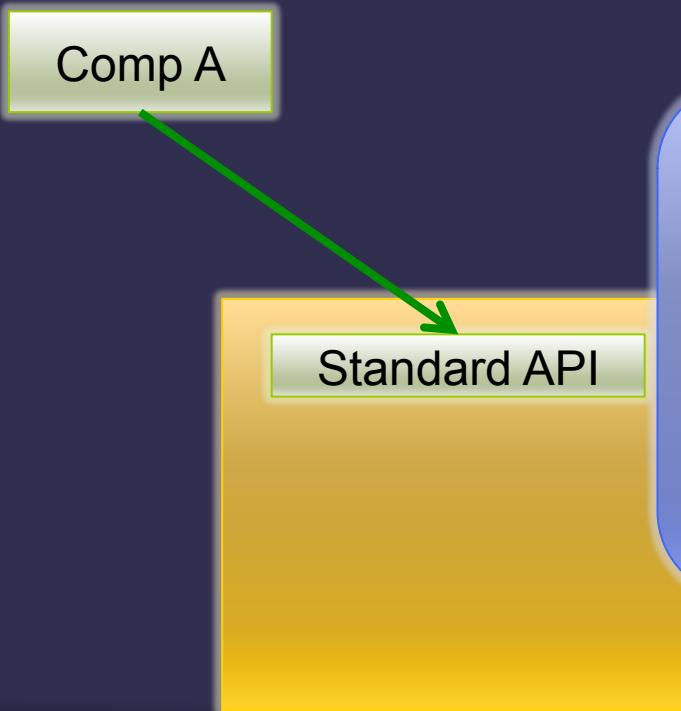
G.U.



CHALMERS

# Optimal architectural decision

- Example:
  - Standard public API

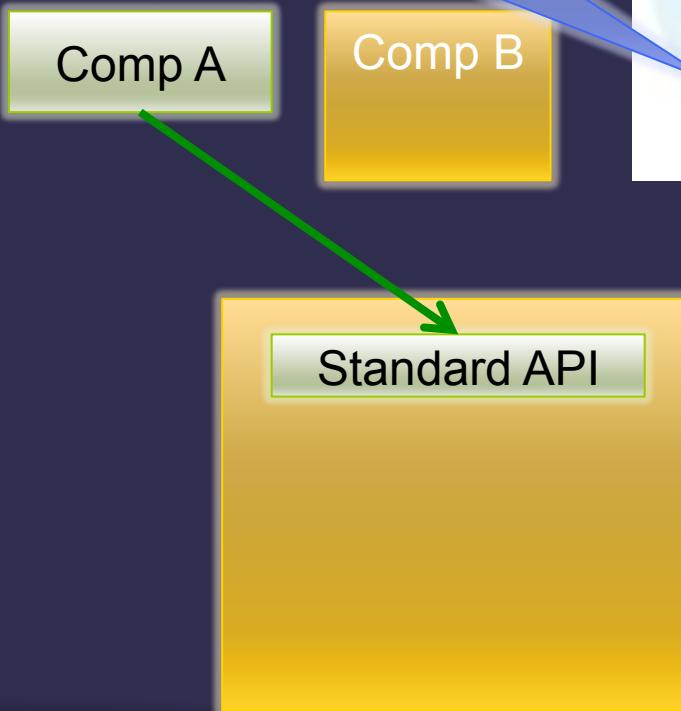


Let's put a standard API here... so later we can update the component independently



# During feature development...

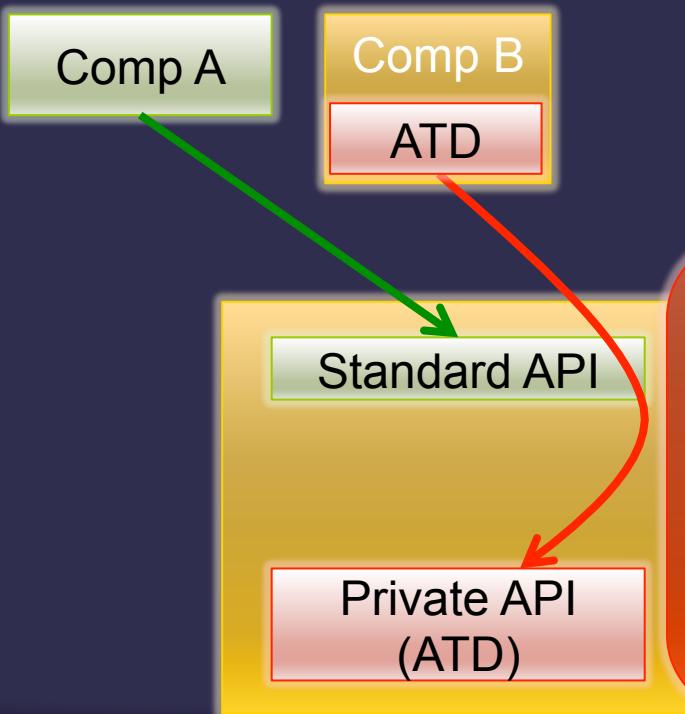
No problem, let's add a component B.  
The teams will use the standard API!



We need these new features! Our competitor is already delivering them!

# ...with fast delivery comes...

- Deliver fast!



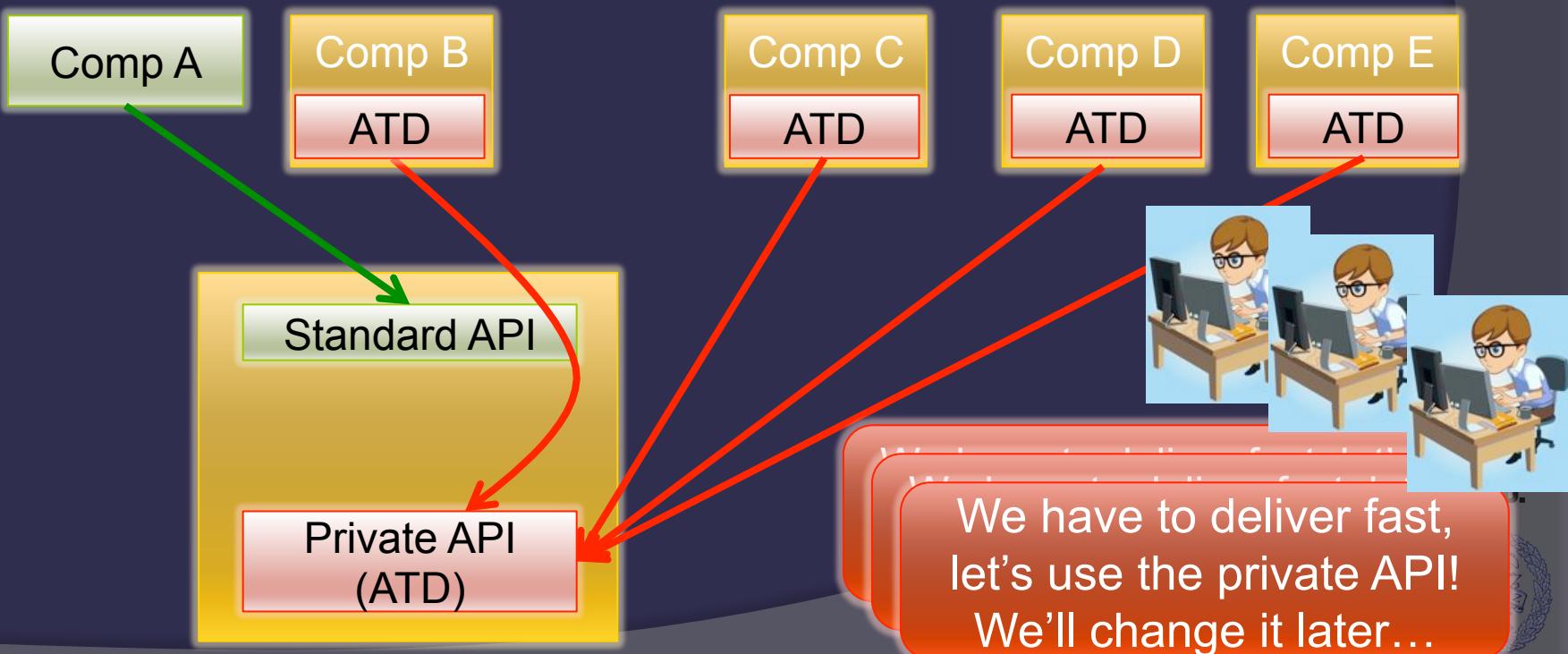
We have to deliver fast, let's use the private API... we'll change it later

We need these new features! Our competitor is already delivering them!

Fast!

# ...the accumulation of sub-optimal decisions...

- The violation is spreading to many components

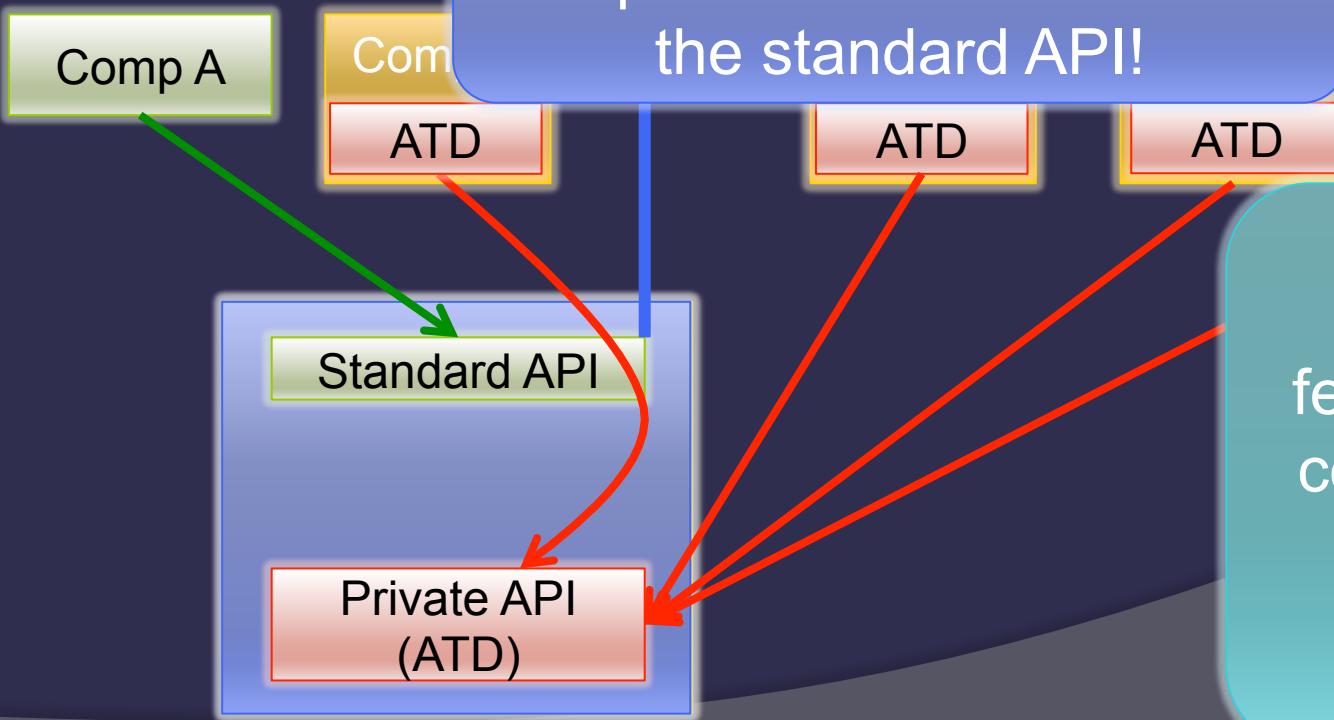


# ...until, one day...

- New requirement



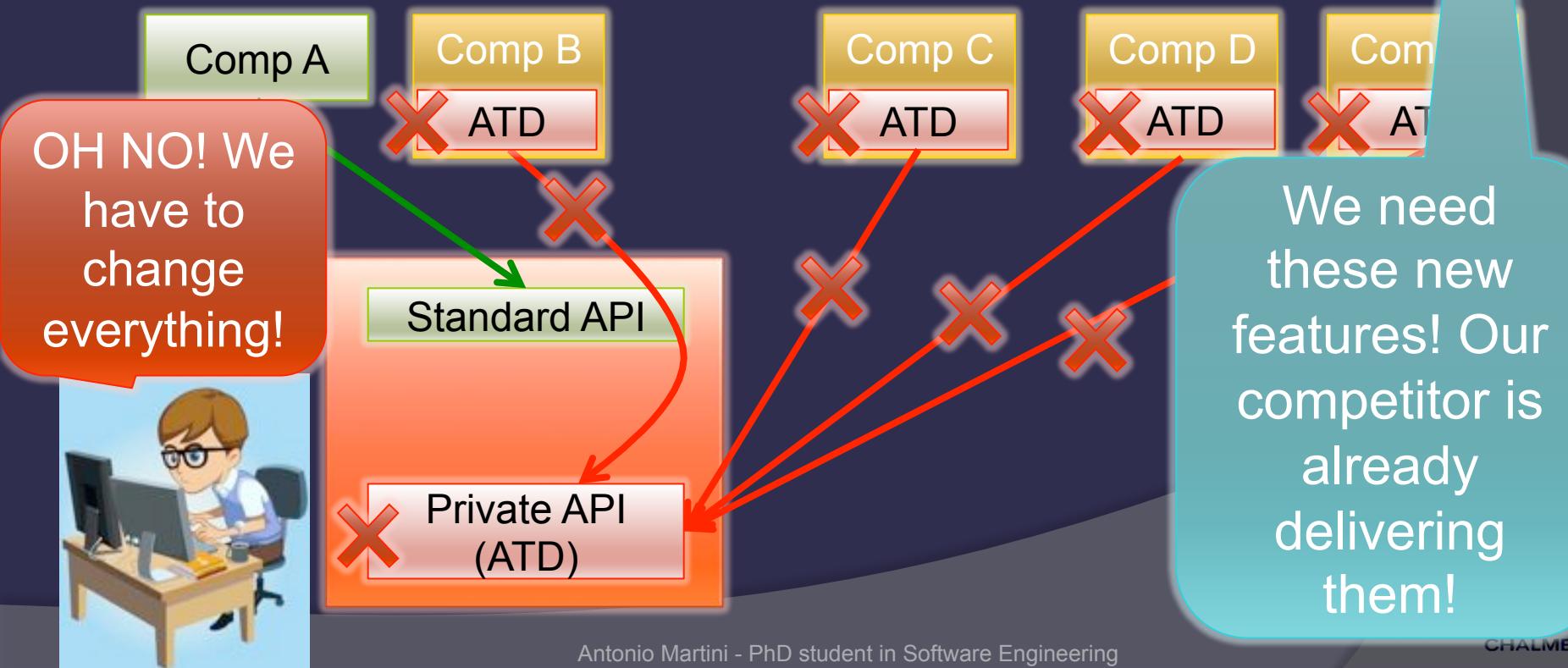
Ok, we can replace this component. The teams used the standard API!



We need these new features! Our competitor is already delivering them!

# ...the development is not fast anymore...

- **Costly** to remove the violation and *difficult to estimate the impact*

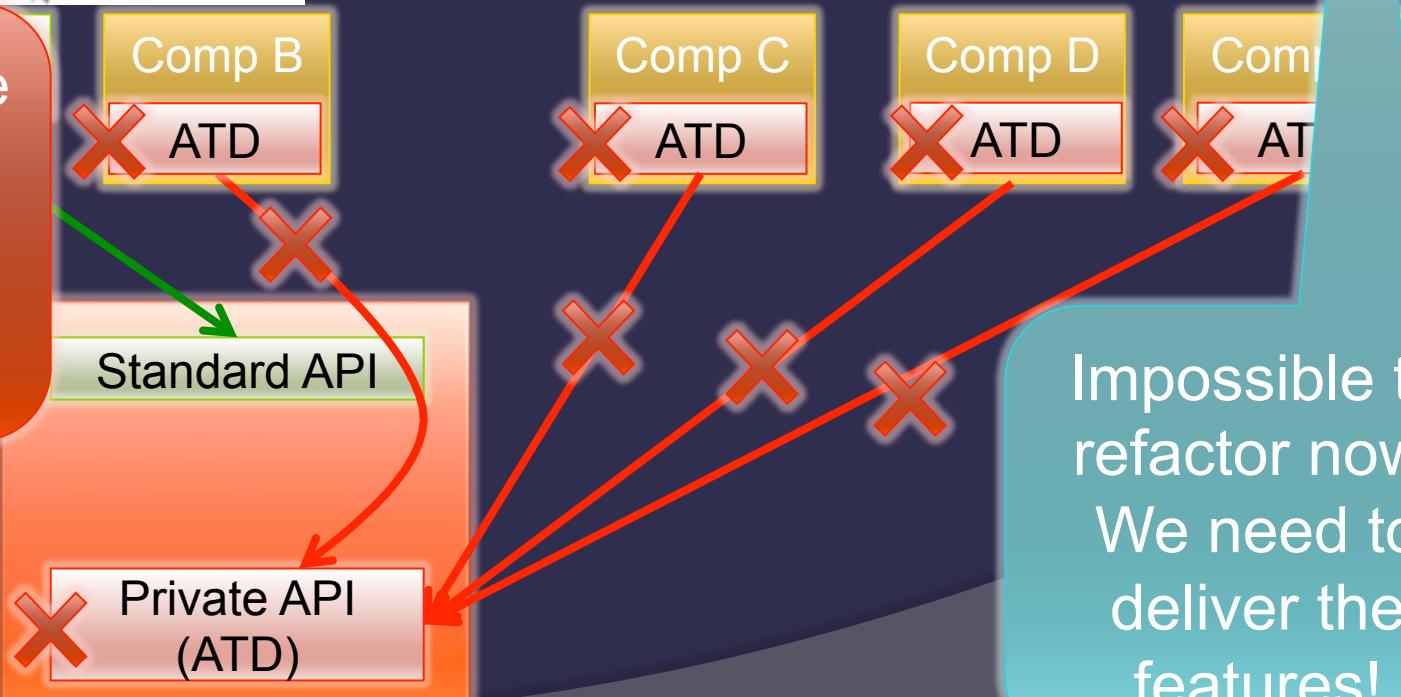


# ...and a crisis starts.



We have to refactor, but we need time...

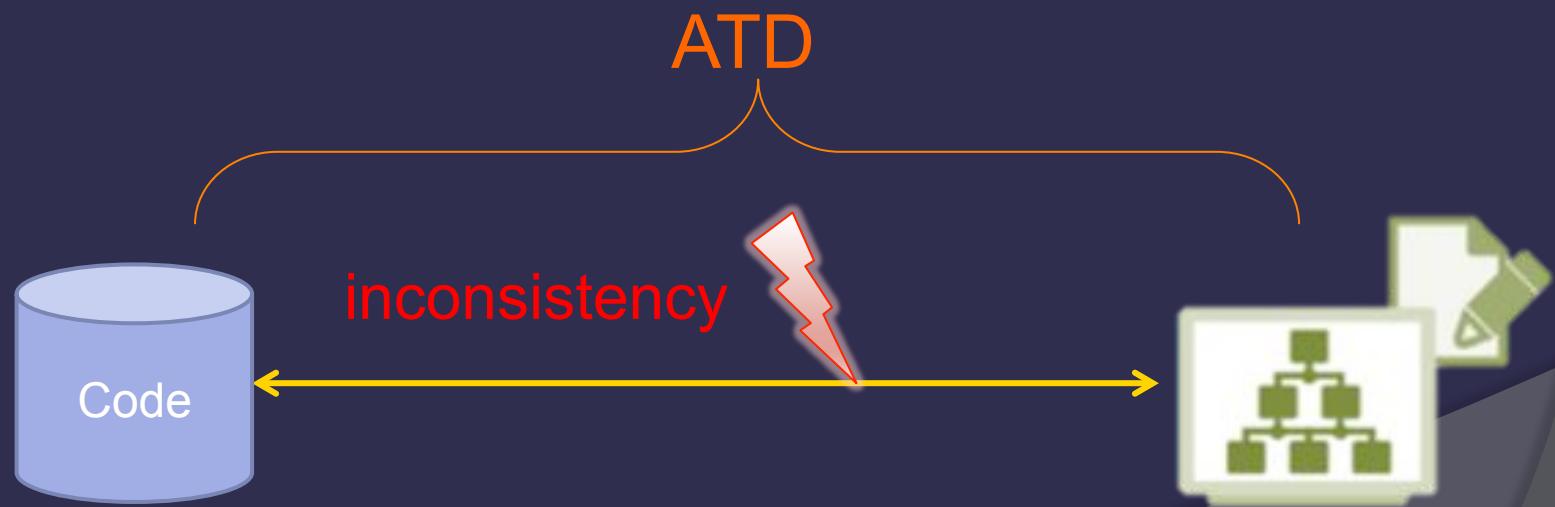
So should we refactor or continuing with other features?



Impossible to refactor now!  
We need to deliver the features!

# Architecture Technical Debt (ATD)

- Inconsistencies (violations) represent the debt between:
  - Current code
  - Desired Architecture



\*Not necessarily completely documented!

# Architecture Technical Debt in a real example

- Non-allowed dependencies = “Taking” the Debt
  - Save time by non-applying the optimal solution
- Cost of removing dependencies = Principal
  - How much does it cost to provide the optimal solution?
- Extra evolution cost

  - Replacing the component

- Other impacts = Interest
  - Increasing principal over time
  - Difficult estimation of refactoring
  - Increased lead-time



G.U.



CHALMERS

# Architecture Technical Debt in a real example

- Non-allowed dependencies
    - Save time by non-applying the optimal solution
  - Cost of removing dependencies
    - How much does it cost to provide the optimal solution?
  - Extra evolution cost
    - Replacing the component
  - Other impacts
    - Increasing principal over time
    - Difficult estimation of refactoring
    - Increased lead-time
- = “Taking” the Debt
- = Principal
- Important to be understood
- = Interest

Managing Architecture Technical Debt: Organization and practices

# Recent Research Results



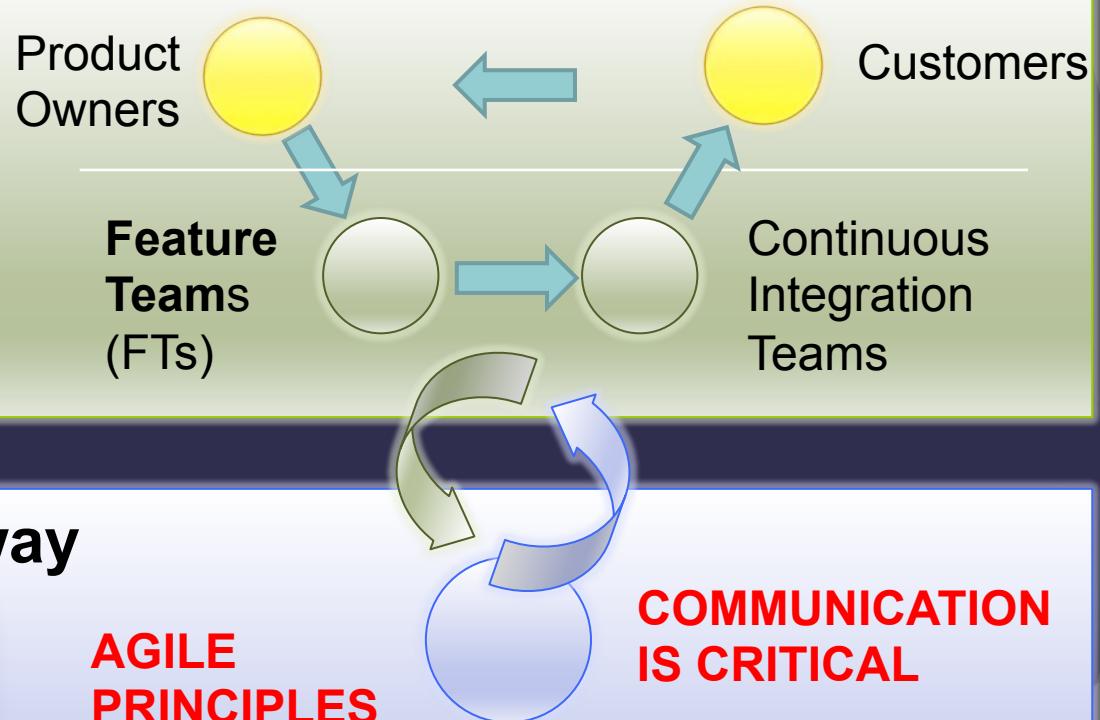
G.U.



CHALMERS

# Research Goal: introduction of Agile Architecturing

## Agile



## Architecture Runway



AGILE  
PRINCIPLES  
CRITICAL

COMMUNICATION  
IS CRITICAL  
ORGANIZATION  
CRITICAL

Agile Architecturing

# Research Results for Organization\*

- CAFFEA framework
  - Architecture Roles
  - Architecture Teams
  - Architecture Practices



## CAFFEA

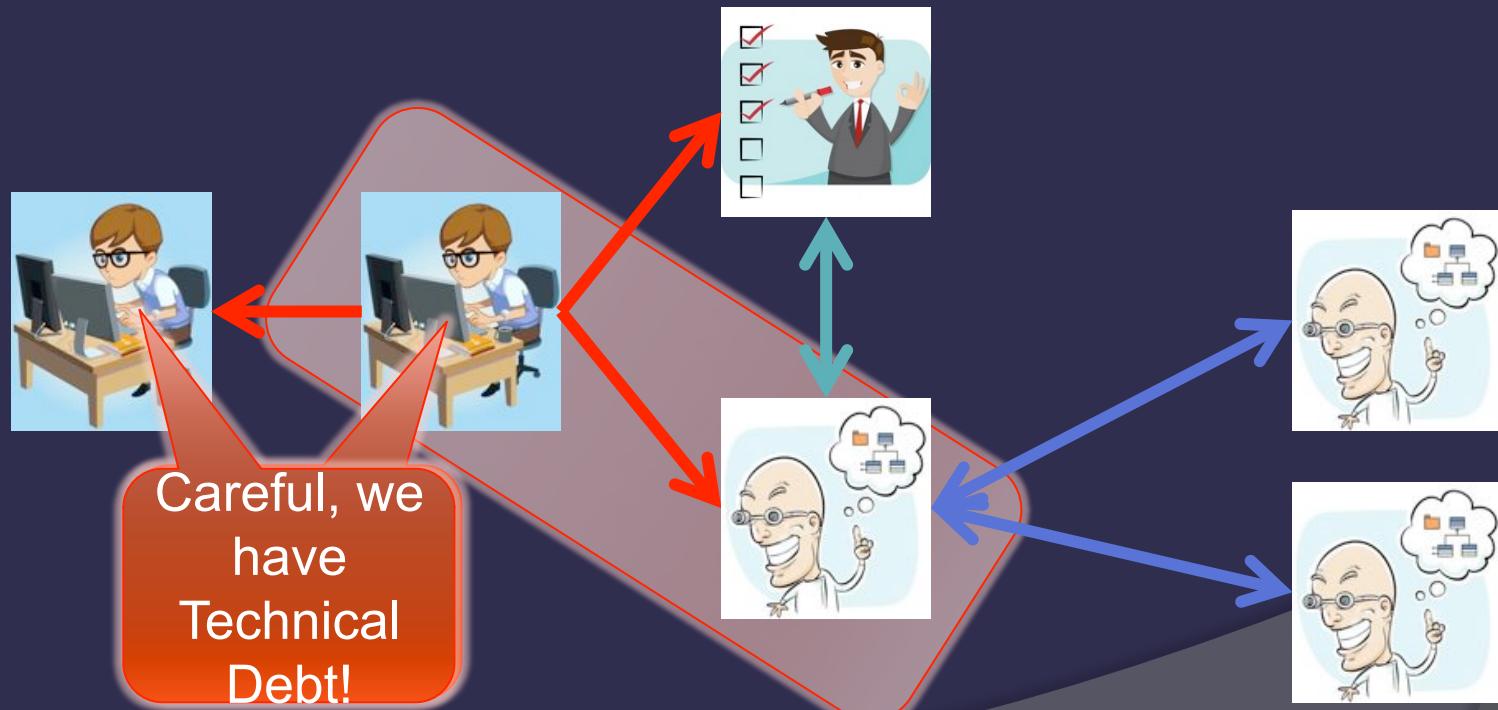
- Governance Architect
- Team Architect



\*A. Martini and J. Bosch, "Towards introducing Agile Architecting in Large Companies: the CAFFEA framework," in XP Conference 2015.

# Research Results for Organization\*

- CAFFEA framework
  - Architecture Roles
  - Architecture Teams
  - Architecture Practices



A. Martini and J. Bosch, "Towards introducing Agile Architecting in Large Companies: the CAFFEA framework," in XP Conference 2015.

Antonio Martini - PhD student in Software Engineering

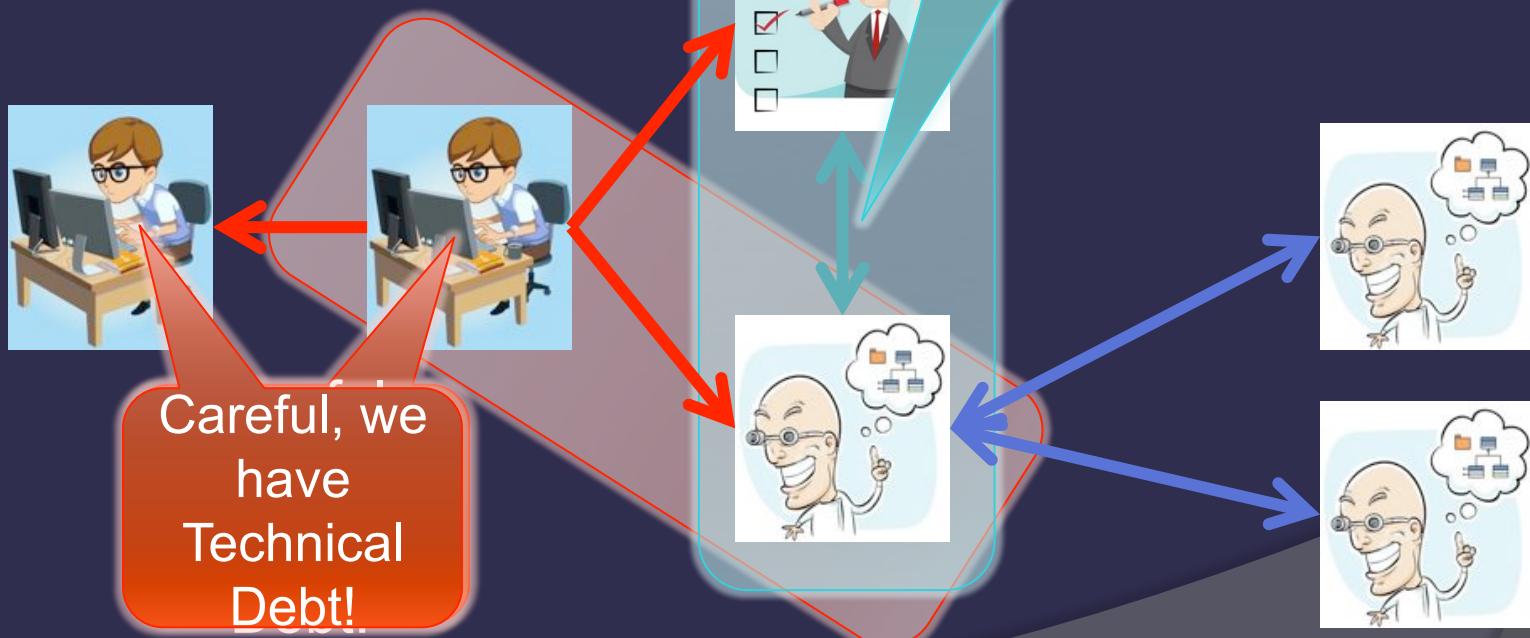


CHALMERS

# Research Results for Organization\*

- CAFFEA framework
  - Architecture Roles
  - Architecture Teams
  - Architecture Practices

What to do next?  
Refactoring or  
features?



A. Martini and J. Bosch, "Towards introducing Agile Architecting in Large Companies: the CAFFEA framework," in XP Conference 2015.

Antonio Martini - PhD student in Software Engineering

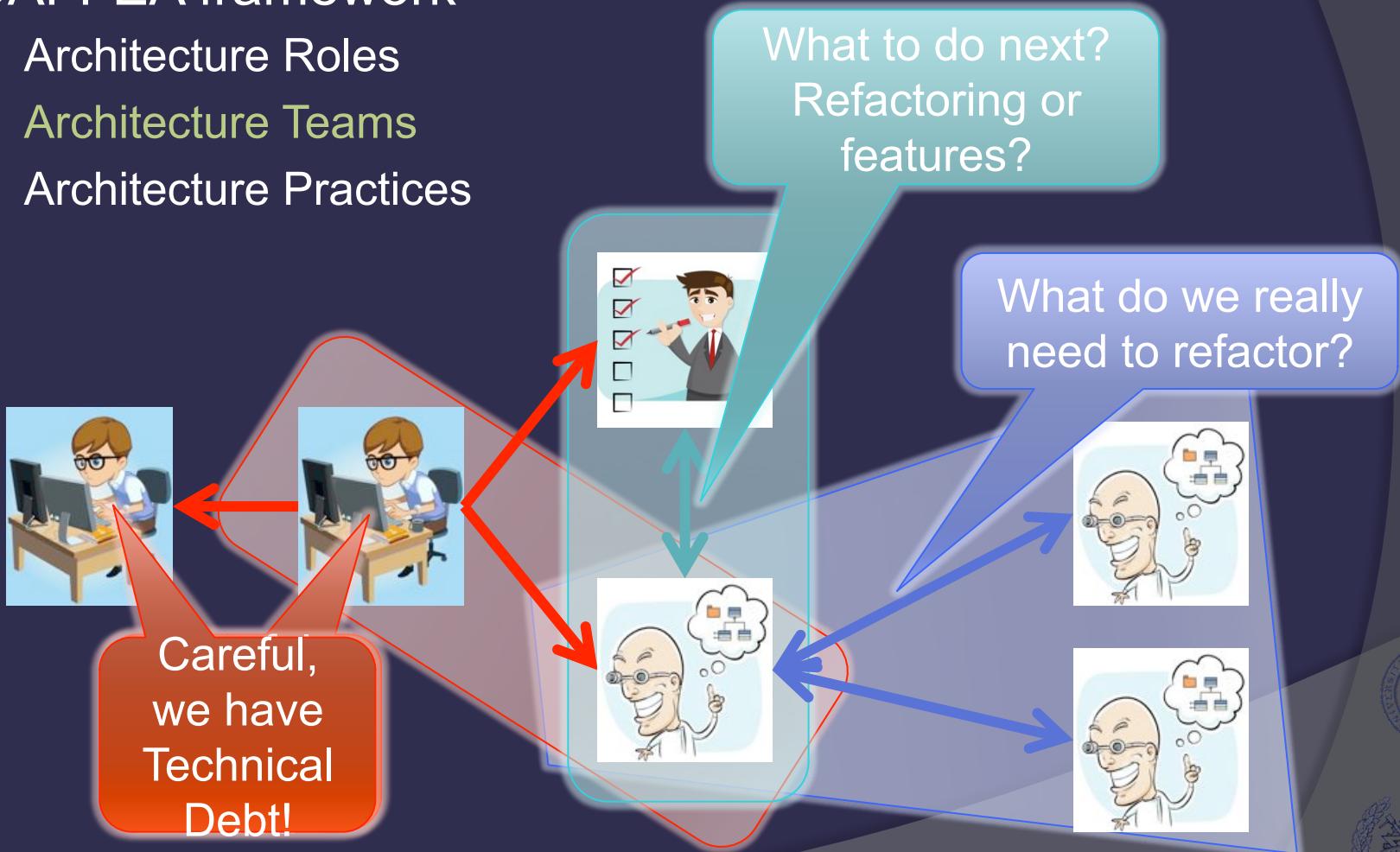


CHALMERS

# Research Results for Organization\*

## ○ CAFFEA framework

- Architecture Roles
- Architecture Teams
- Architecture Practices



A. Martini and J. Bosch, "Towards introducing Agile Architecting in Large Companies: the CAFFEA framework," in XP Conference 2015.

Antonio Martini - PhD student in Software Engineering



CHALMERS

# Practices findings\*

- Practices to be more and/or better handled:

- Architecture Education
- Architecture Decision
- Architecture Consistency
- Architecture Prioritization
- Tool Decision

Architectural Technical  
Debt Management

Architecture Team (AT)



Governance Team (GT)



# Questions?

# Comments?

- References:

- To know more about the previous project results, for comments and questions and for thesis opportunities on the subject contact:
  - [antonio.martini@chalmers.se](mailto:antonio.martini@chalmers.se)



G.U.



CHALMERS