

The role of the architect in an agile project setting

Dominik Rose
Alexander Mayer
InterFace AG

11.12.2013

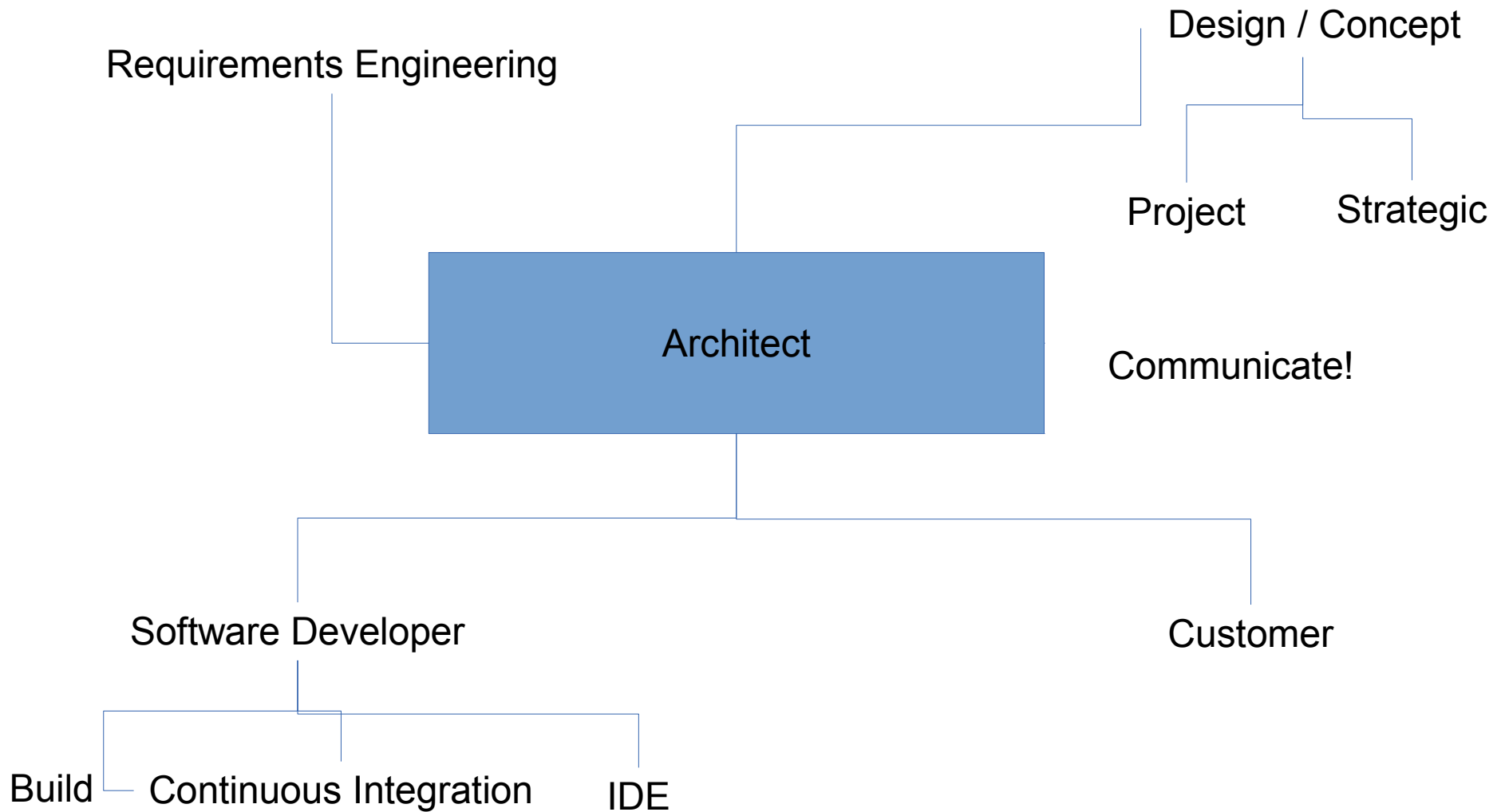


What is architecture?

“The software architecture of a program or computing system is the **structure or structures of the system**,

which comprise **software elements**,
the **externally visible properties** of those elements,
and the **relationships** among them.”

What is the role of the architect?



There are good reasons people started to think
about Agile...

...wait – what exactly is Agile?

Agile Manifesto (2001)

„We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

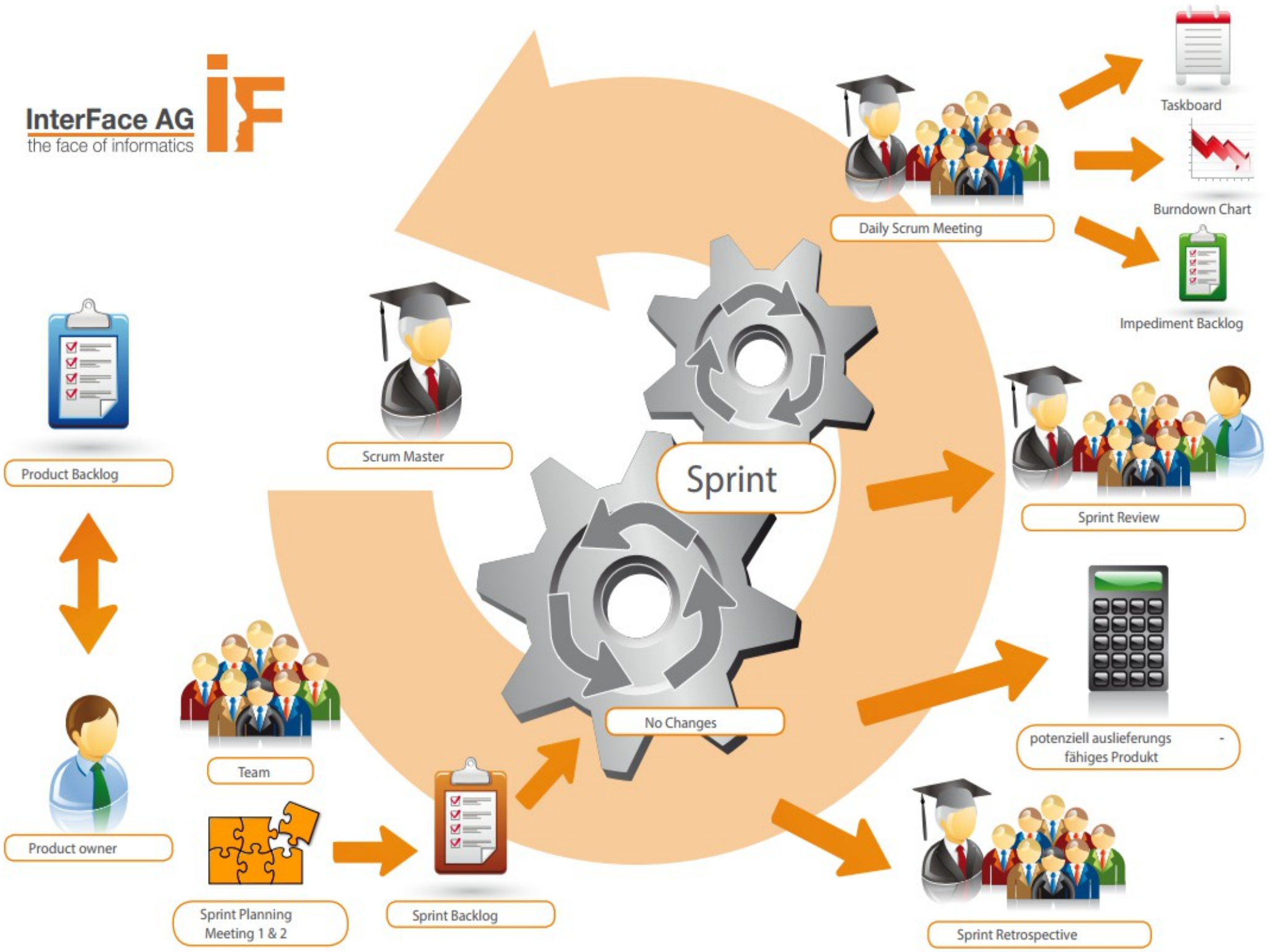
Individuals and interactions > processes and tools (self-organizing teams)

Working software > comprehensive documentation (working software is delivered frequently)

Customer collaboration > over contract negotiation (close, daily cooperation between business people and developers)

Responding to change > following a plan (welcome changing requirements)

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



Myth #1: Your system is not gonna change

Myth #2: Agile is chaos

Myth #3: Agile don't need documentation

Myth #4: Architecture is big-upfront design

Myth #5: Agile projects don't need architecture

Myth #6: Agile projects don't need an architect

Quiz: Which myth is wrong?

Myth #1: Your system is not gonna change

ALWAYS WRONG

Myth #2: Agile is chaos

SOMETIMES IT IS, AND SOMETIMES IT IS NECESSARY

Myth #3: Agile don't need documentation

IT DEPENDS

Myth #4: Architecture is big-upfront design

IT DEPENDS

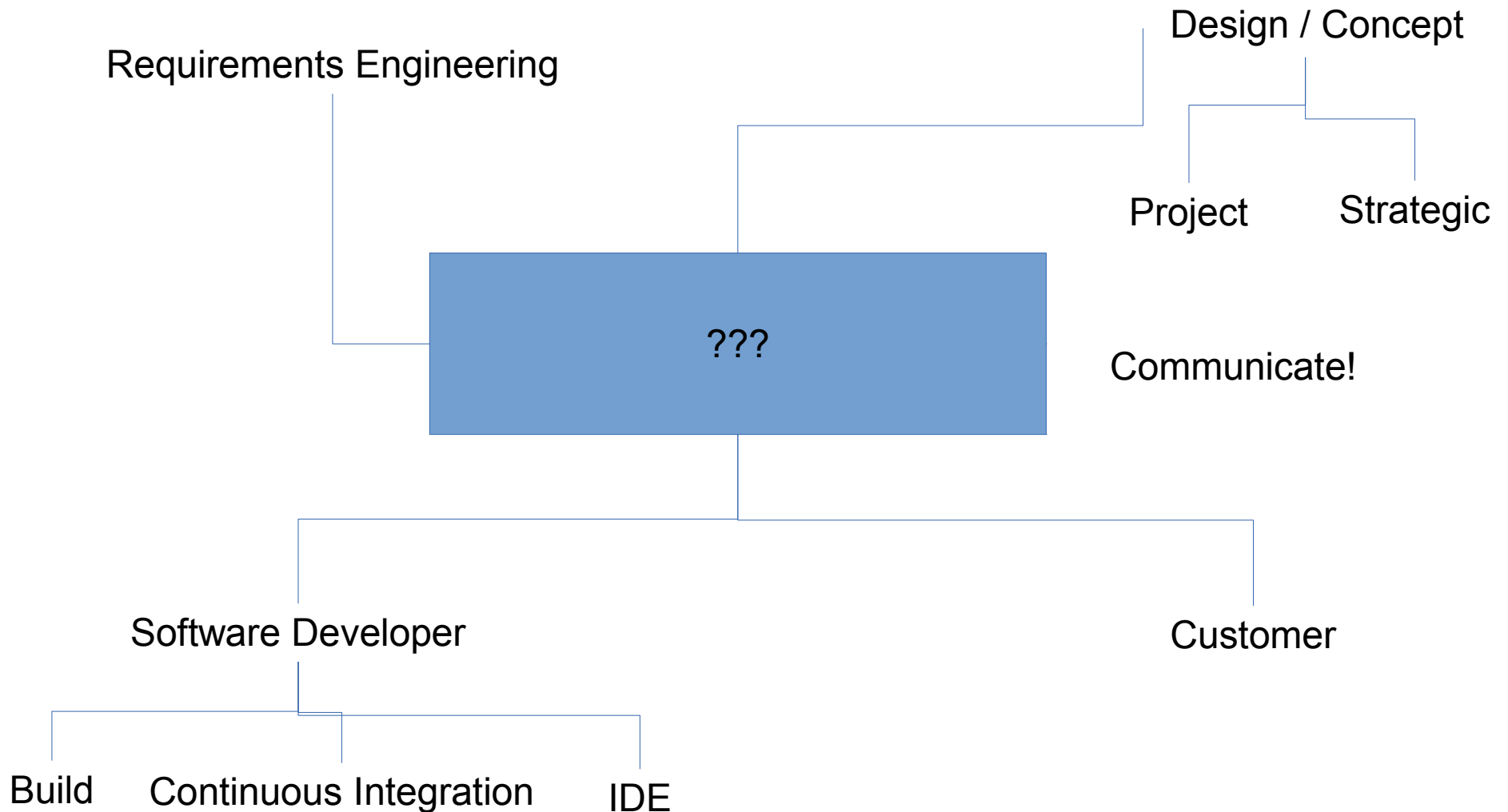
Myth #5: Agile projects don't need architecture

ALWAYS WRONG

Myth #6: Agile projects don't need an architect

SOME TRUTH IN IT

Who is the architect then?



a) The team is the architect

- Architectural decisions as part of documentation
- Team discovers architectural conflicts with requirements, Scrum-Master communicates
- But:
 - Does it scale?
 - Strategic decisions?
 - Project smells?
 - Big picture?
 - Nobody knows all!
 - Experienced developers?

b) Architect in an agile project

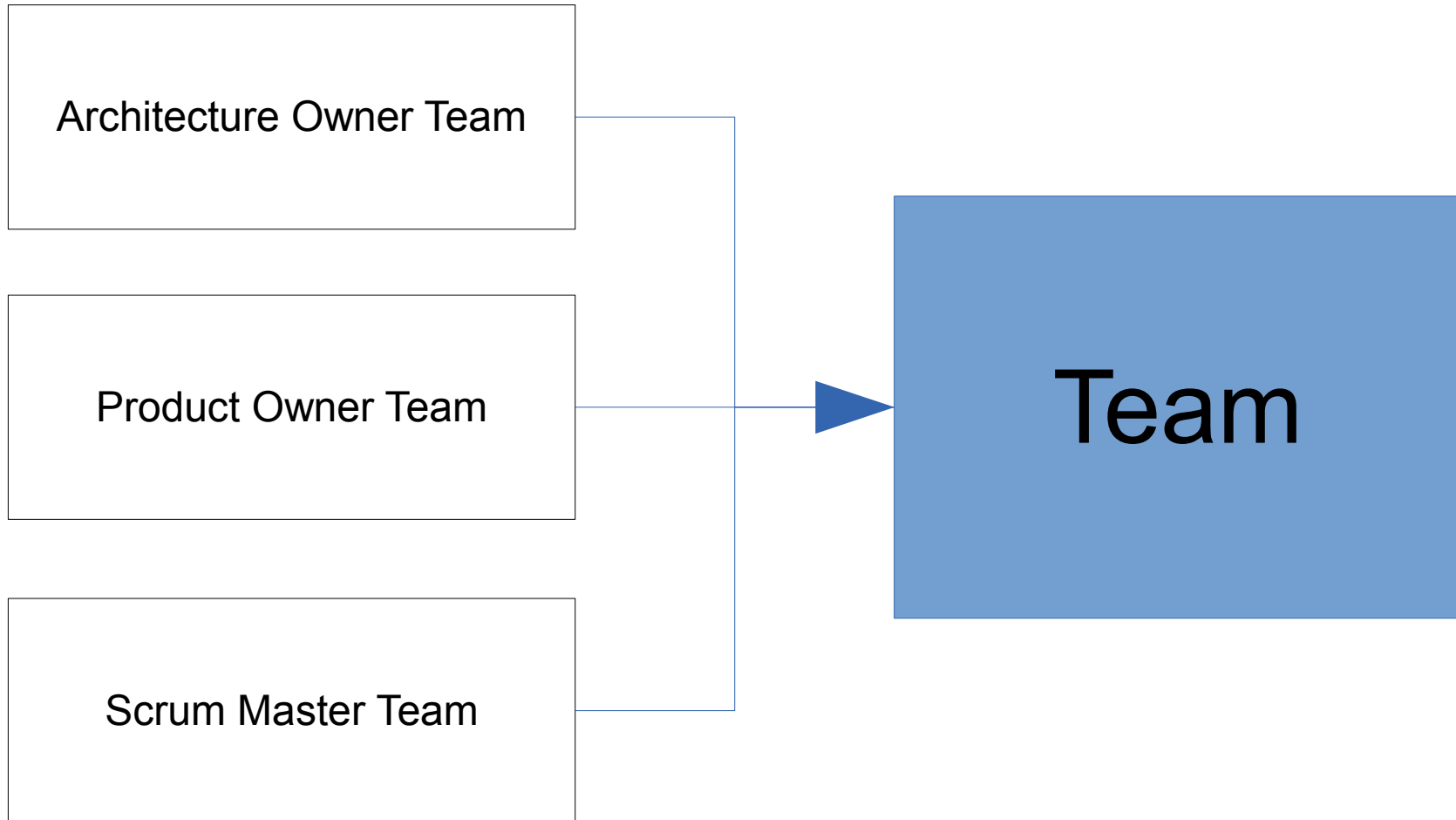
- Part of the team
- Hands on
- Non functional requirements
- Identifying risks and how to solve them
- Strategic / technological decisions
- Coordinate and communicate with stakeholders

c) Architecture Owner

Scott W. Amber:

- Facilitates the evolution of the software
 - Communication and Discussion with the team!
- Transitioning architectural knowledge to other team members
- Architectural spikes (risks)
- Organizational / Technical Mentoring
- Breaking decision deadlocks
- Code review

Agile at scale



c) Architecture Owner

- Usually coordination between architects needed
- Architecture Owner Team
- Scrum-Master invites Architect to the team
- Architects form separate Scrum Team
- Conflicts with Product Owner?

Goals of an agile architecture

- KISS
(„Everything Should be Made as Simple as Possible, but not simpler“)
- Modular
- Readable
- Documented
- Testable
- Don't constrain the future, be ready for future changes – no over engineering!
- Continuously refactored

Simple is hard

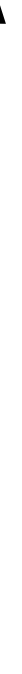
„Simple solution“

„Fits into architecture“

„Documentated“

„Tests successful“

„Build successful“



„write once, read never“

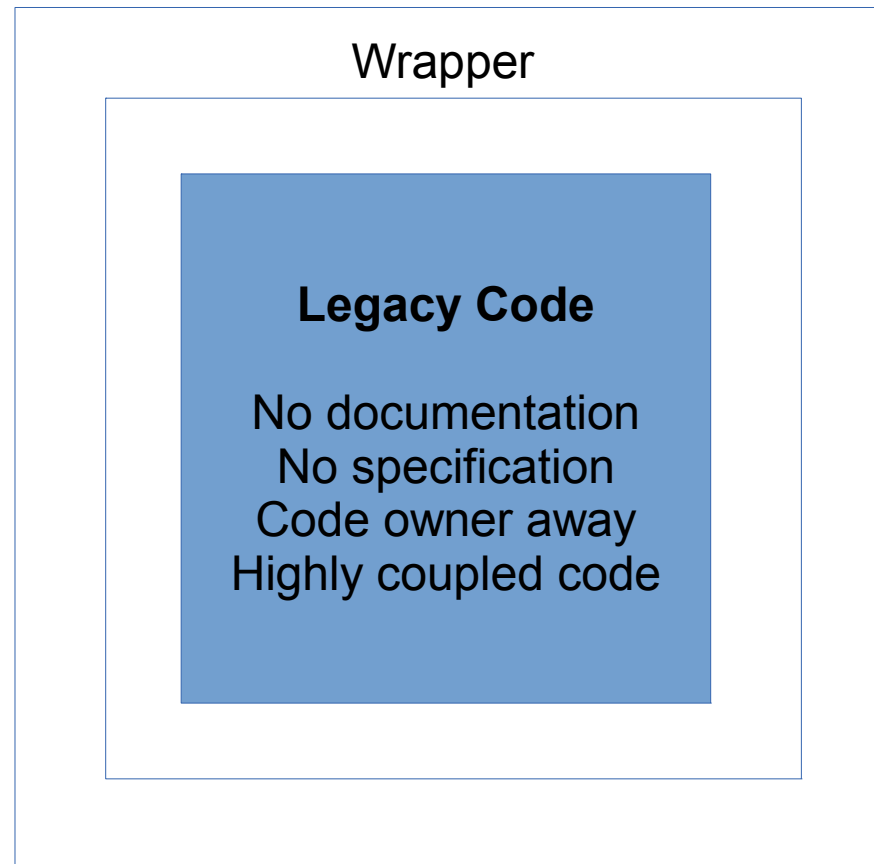
Quality

- Performance
- Security
- Reliability
- Testability
- Usability
- Modifiability

Always trade-offs, always sensitivity points

=> Needs to be documented

Architecture turned into stone



Keep your code agile!

So what is Agile Architecture?

1. Important decisions - Those that are expensive!
2. Trade-offs
3. Sensitivity points
4. Risks

=> Document not what the system does, but why!

=> Communicate!

How to learn agile architecture?

1. Practice
2. Practice
3. Practice
4. Practice
5. Practice



<http://en.wikipedia.org/wiki/File:laido2.jpg>

(e.g. Architectural Katas: <http://archkatas.herokuapp.com>)

Client wants to create websites to follow Internet trends--as a new trend is identified, create a website following it and highlighting it and allowing users to interact with it

Users: millions+ readers, thousands+ posters, dozens admins

Requirements:

- high SEO;
- easy for users to add content;
- easily mashable/clickable;
- reject inappropriate content;
- easy trend analysis;
- user forums;
- user moderators;
- ubiquitous accessibility;
- easy admin "reach"/accessibility

Who are the actors?

What are the sensitive points?

What are the risks?

What are the trade-offs?

Summary

Embrace change!

Don't be dogmatic!

Every team member should have architectural skills.

Architects are needed for the big picture.

A good software architect is an agile architect.

Thank you!

alexander.mayer@interface-ag.de
dominik.rose@interface-ag.de