

Catalysts

SW-Architecture Root Causes

Dr. Christoph Steindl
DI Christian Federspiel
(steindl | federspiel) @catalysts.cc

What drives Architecture?

Catalysts

- Team size (number of people), knowledge of members
- Project size (duration and Mio €) – time and money
- Criticality (potential loss of money or life)
- Complexity
- Non Functional Requirements (NFRs) like availability, performance, scalability; but also functional aspects
- Experience (of the involved people, previous problems, war stories) and education
- Standards (within the organization, from external, industry,...)
- Risk willingness of customer
- Eagerness of consumers (how many installation/time? 1/week or 1/year?)
- Technology, legacy systems, existing systems management and operations skills
- Architect fame (If solution is easy, problem was easy!)
- Available data material / test material
- Financing Model
- Organization (see Conway's law)
- Process (Waterfall → complex architecture; Agile → simplistic, simple architecture)

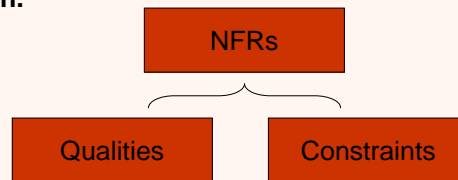
© Catalysts, 2006

Driving Forces

2

What drives Architecture? Catalysts

- Business and IT
- Costs for Development, Operations, and Maintenance
- Non-Functional Requirements (or NFRs) define the qualities of the system and the constraints under which the system must be built.
 - **Qualities** define the expectations and characteristics the system should support.
 - **Constraints** are limitations or specifications imposed upon a solution.



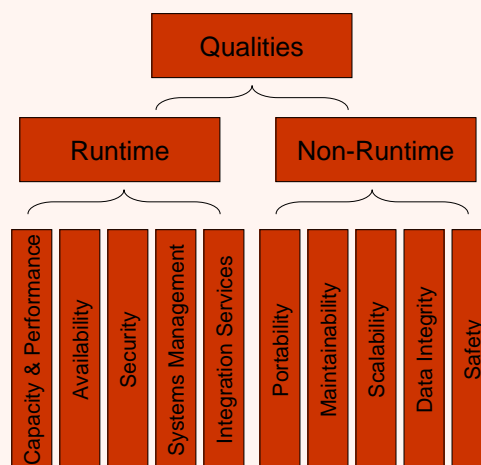
© Catalysts, 2006

Driving Forces

3

Qualities Catalysts

- **Runtime qualities** are expressed by Service Level Requirements.
- **Non-Runtime qualities** might be related to the development, maintenance, or operational concerns that are not expressed at runtime.



© Catalysts, 2006

Driving Forces

4

Constraints Catalysts

- The **business aspects** of the project, customer's business environment or IT organization that influence the architecture
- The **technical environment** that the system needs to operate within

© Catalysts, 2006 Driving Forces 5

Uncertainty Catalysts

Four Types of Project

| | | |
|-----------------|-------------------|-------------------|
| Don't Know What | TECHNO | WICKED |
| Know What | TAME | QUEST |
| | Know How | Don't Know How |

© Catalysts, 2006 Driving Forces 6

Change

Catalysts

- Competitors impose changes
- Business changes
- Clients want new features not compatible with architecture
 - New functional requirements
 - New non-functional requirements
- Middleware changes (database server, application server, portal server)
- Operating system changes
- Systems management changes
- Data access layer changes (JDBC, DAO, EJB, Hibernate)
- Interfaces to external systems, legacy systems
- Specification changes (EJB 2.1 → EJB 3.0)
- Language changes (Java JDK 1.4.2 → 1.5 / 5.0)

© Catalysts, 2006

Driving Forces

7

Traditional assumptions

Catalysts

- Certain aspects of an architecture cannot be changed later (or it is very difficult to change them).
 - Multi... language / currency / nationality / client support
 - Distribution and scalability
 - (Distributed) transactions
 - Security and safety
 - Data base schemes
 - Web-/Swing-/SOAP- UI
 - ...

© Catalysts, 2006

Driving Forces

8

What is nevertheless changed after the end of a project

Catalysts

- Introduction of access rights and roles
- Conversion from fat client to web client
- Fragmentation of a monolithic application into an Eclipse plug-in
- Addition of a layer ⇔ combination of layers
- Distribution among servers / remoting ⇔ re-collocation of distributed components
- Introduction or change of logging, exception handling or input validation
- Conversion from horizontal layers into vertical layers
- Conversion of layers into components
- Conversion of components in services (or vice versa)
- Extension of an object-relational access layer with persistency that spans transactions
- Exchange of the underlying database
- Refactoring of the database schema
- Porting to another operating system
- Conversion to OSGi (Eclipse 3.x)

© Catalysts, 2006

Driving Forces

9

So we have a lot of drivers... Catalysts

- Simply knowing them can be valuable.
- But how can we know where to focus on?
- How do we know what is really important?
 - What should be done?
 - What should be done first?
 - What should be delayed?
- We have to **analyze the drivers**, their **associated risks** and **propose an approach**, be able to **defend it** and **judge whether it was successful**.

© Catalysts, 2006

Driving Forces

10

Is there a "one size fits all"? Catalysts

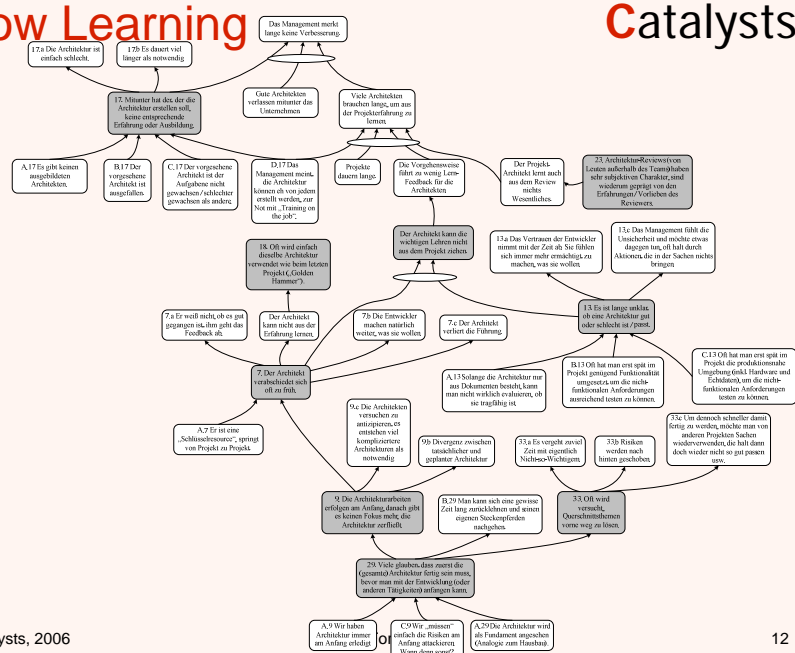
- Is there one approach that fits all situations?
- Can there be one such approach?
- We have seen in our experience that
 - Projects may fail even if there is a lot of architecture work
 - Project may succeed even if there is little architecture work
- We have brainstormed the negative effects that we have experienced.
- We have analyzed their reasons, asked why they came to exist.
- We have found some interesting points.

© Catalysts, 2006

Driving Forces

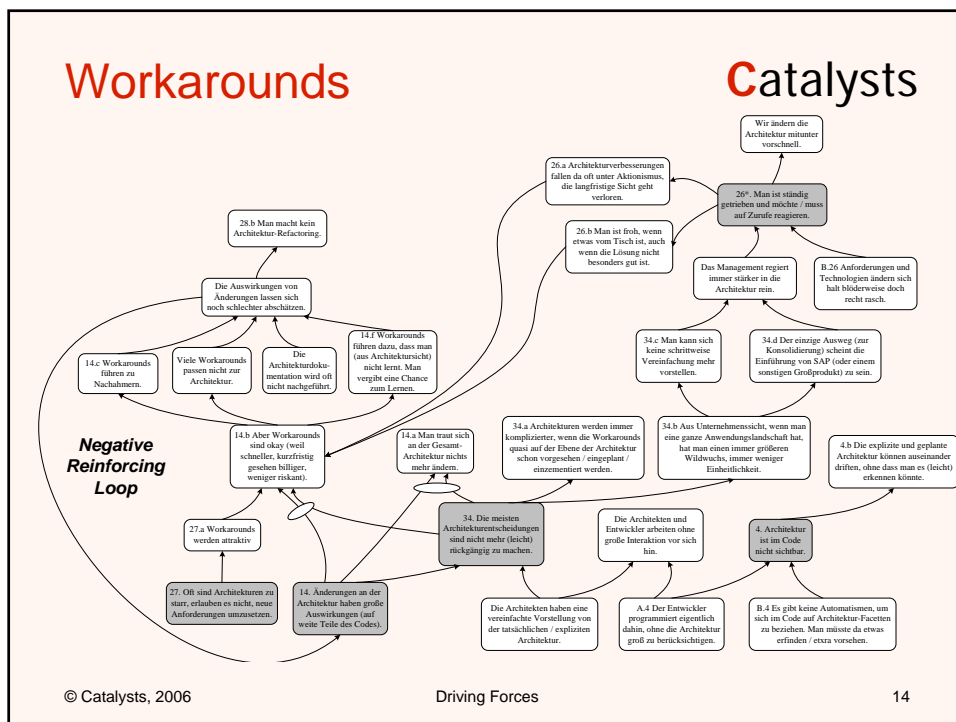
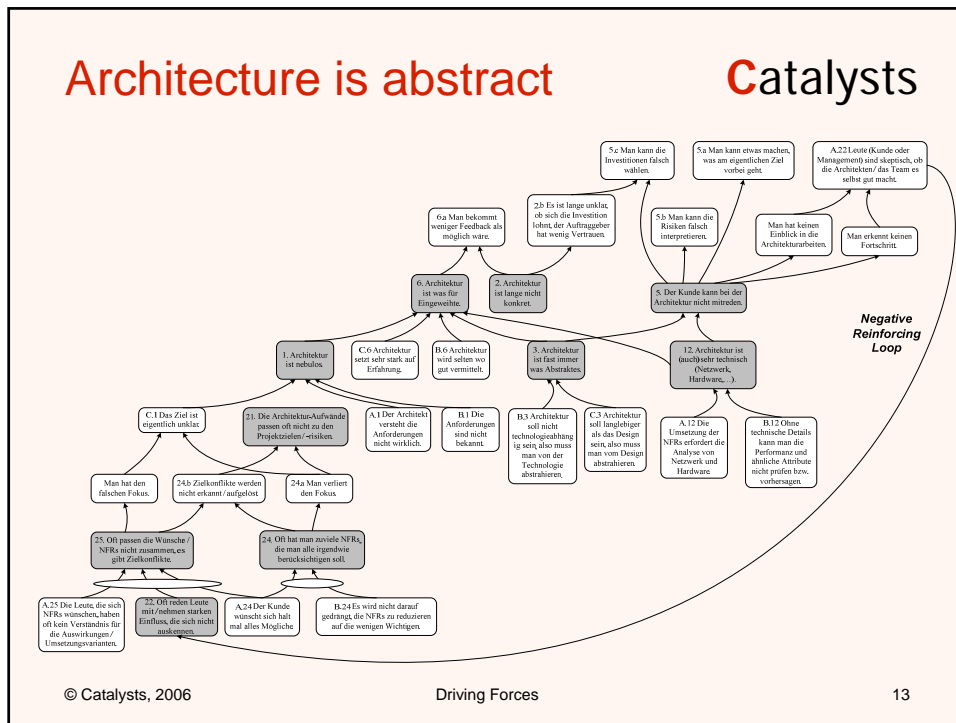
11

Slow Learning Catalysts



© Catalysts, 2006

12



Exercise: Root Causes

Catalysts

- We will take a few minutes to brainstorm problems with architecture work in a structured way, by listing
 - under “**Negatives**” all the indications you can see that architecture work isn’t performing as it should or is not giving the results / benefits / payback that it should. Number the “Negatives” sequentially.
 - under “**Why?**” all the reasons why the Negative is bad. Number the “whys” to correspond with each “negative.” Use numbers and letters (e.g. 1.a, 1.b, 1.c for the whys for the negative 1.)
 - under “**Causes**” all the reasons why the “Negative” exists. Number the “causes” to correspond with each “negative.” Use letters and numbers to distinguish the “causes” from the “whys” (e.g. A.1, B.1, C.1 for the causes of negative 1.)

| Causes | Negatives | Why? |
|--|------------------------------|--|
| A.1 | 1. | 1.a |
| A.1 The architect does not really understand the requirements. B.1 Requirements are unknown C.1 The goal is rather vague | 1. Architecture is nebulous. | 1.a Discussing the architecture is hard. 1.b Discussion is limited to inaugurated experts / insiders (0-1 people). 1.c One can talk about something for a long time without recognizing misunderstandings. |

© Catalysts, 2006

Driving Forces

15

Exercise: Root Causes (continued)

Catalysts

- We do the initial brainstorming of negatives in an anonymous way.
- We discuss the negatives and challenge their impact as a group.
- You develop the causes in small working groups.
- Each working group presents their work.

- We gather the findings and scrutinize the logic overnight.
- We try to identify root causes for the negative effects.
- We will present tomorrow what you have found.

© Catalysts, 2006

Driving Forces

16

Exercise: Root Causes (continued)

Catalysts

- For scrutinizing the logic, please ask the following questions repeatedly:
 1. Is the *statement clear*? Is the causal *connection clear*?
 2. Are all statements *complete sentences*? Do all statements convey *only one idea*? Is each statement *valid*?
 3. Does causal *connection really exist*?
 4. Is the *cause sufficient* to produce the effect? Are all *contributing causes really needed*?
 5. Can any *other causes independently* produce the same effect?
 6. Are the cause and effect actually *reversed*?
 7. Is the cause *intangible*? If so, are any *other effects* observable to verify it?

Appendix

Catalysts

Catalysts

- Founded in 2005
- Initial committer to the Eclipse Process Framework (EPF, <http://www.eclipse.org/proposals/beam/>)
- Corporate Member of the AgileAlliance
- We are determined to improve your productivity, flexibility, and resilience.

Catalysts



Christoph Steindl

- Founder of Catalysts in 2005
- Senior IT Architect and Method Exponent with IBM (2000 - 2005)
- Certified ScrumMaster
- Member of the ScrumAlliance
- Member of the AgileAlliance
- NLP Business Practitioner
- Invited Lecturer at the Johannes Kepler University in Linz and at the University of Applied Sciences in Hagenberg
- Studied Computer Science (Dr. techn.) and Mechatronics (Dipl.-Ing.)

Catalysts



Christoph Steindl – IBM

Catalysts

- I jumped right into a tough project with a tight deadline in 2000.
- I became a team leader soon and a Method Exponent.
- I got certified as an IT Architect in 2002.
- I worked on large and complex projects balancing agility with discipline.
- I gave talks at IBM-internal conferences about topics like interactive portal applications, Use Cases, IP telephony, distributed agile, lean software development, etc.
- I became a lead instructor for Use Case Modeling and Architectural Thinking.
- I was elected to join the Technical Experts Council of the Central Region in 2004.
- I became a member of the core team of the world-wide community Agile@IBM in 2004.
- I gave a talk about estimation in agile projects at a conference of the IBM Academy in 2005.
- I built my reputation first in Linz, then in the Central Region (D/A/CH), and finally in Vienna, with increasing world-wide visibility.

Christoph Steindl – Lean

Catalysts

- Mary and Tom Poppendieck release their book [Lean Software Development](#) in 2003.
- I read the book and got in contact with the Poppendiecks.
- I created an [extended excerpt](#) and spread the word within IBM at internal conferences.
- I gave a talk at the [JAOO conference](#) in Aarhus in 2005 about *Applications of Lean Software Development*.
- I reviewed their new book [Implementing Lean Software Development](#).
- I continued to read books on [Lean Thinking](#), [Lean Product Development](#), [Lean Solutions](#), [The Toyota Way](#), etc.
- I combined my interest areas [Lean](#) – [Mindful](#) – [Agile](#) for the consulting business of Catalysts.

Christoph Steindl – Agile Catalysts

- The [C3 \(Chrysler Comprehensive Compensation\) project](#) was started in 1995.
 - Kent Beck joined in 1996, brought in Ron Jeffries.
 - They installed what is now known as Extreme Programming.
- I listened to Kent Beck's talk about Extreme Programming at the European Software Engineering Conference (ESEC) in 1999.
- I spread the word and started to apply it.
- 17 renowned experts signed the [Agile Manifesto](#) in February 2001.
- I signed the Agile Manifesto in [April 2003](#).
- I balanced agility with discipline in my projects at IBM from 2000 to 2005.
- I gave talks about agile software development and various aspects thereof at IBM internal conferences.
- I gave public seminars and lectures at the University Linz and FH Hagenberg about *Scrum*, *Agile Project Management*, *Test-Driven Development*.
- I became a Certified ScrumMaster in 2004.
- I created the approach Distributed Agile to work in an agile way with distributed teams.
- I became a core member of the world-wide community "Agile@IBM" in 2004.
- I published an IBM-internal article on how to prevent troubled projects by using agile practices.
- I gave a talk about *From Agile Software Development to Agile Businesses* at the EuroMicro Conference in 2005.

Christoph Steindl – Testing Catalysts

- I've been giving public seminars, talks, and lectures at the University Linz and FH Hagenberg about *Testing of Software Systems* and *Test-Driven Development* regularly since 1999.
- I grewed my network with testing experts.
- I co-authored the chapter on *Testing Web Applications* for the book *Web Engineering* together with Rudolf Ramler and Josef Altmann.
- I gave a talk about *Traditional and Agile Testing* at 4th Annual Workshop on Teaching Software Testing in 2005.
- I gave a talk about *Critical Aspects of Testing during the Software Lifecycle* at the First IIR Topical Conference on Software Testing in 2006.

Christian Federspiel

Catalysts

- Joined Catalysts in 2006
- Leader of Product Development with Siemens-VAI
- Project Manager with Siemens-VAI
- Coach (trained by Dr. Christoph Haug, CSI)
- Member of the AgileAlliance
- Studied Computer Science (Dipl.-Ing.)