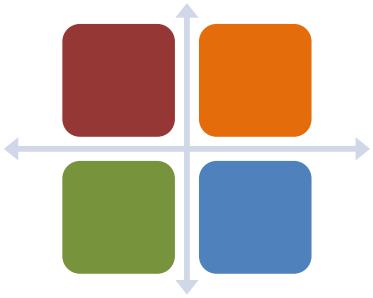


Software Architecture: Perspectives on a maturing discipline

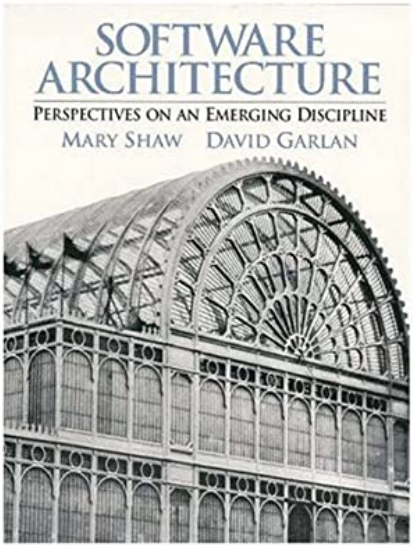
Philippe Kruchten

1

Perspectives on an emerging discipline

Mary Shaw
David Garlan

1996



2

Early history of software architecture

- NATO conference (1969)
- Box & arrows (1960s-1980s)
- Views & viewpoints (1990s-2000)
- ADLs (1980s-2000s)
- Architectural design methods (1990s-2000s)
- Standards, reference architectures (1995-…)
- Architectural design decisions (2004-…)

Kruchten Obbink Stafford 2006

3

My own trajectory

- 1984: I became “System Architect” at Alcatel
  - Butler Lampson (1983) Hints for computer system design
  - John Mills (1985) Pragmatic view of the system architect
- 1987: Technical consultant at Rational Software
  - Exposed to a wide range of large systems, all over the world
  - Internal community of practice
  - External software architecture community

4

Software Architecture Tutorial ca. 1998
Outline

1. **Semantics** – definition: what is it?
2. **Representation** – how does it look like?
3. **When** – in the lifecycle is it done?
4. **Who** – who's doing it?
5. **Process** – what is the process / method to design a software architecture ? (any tool support?)

5

Today's 5 viewpoints

1. **Semantics** – definition: what is it?
2. **Representation** – how does it look like?
3. **When** – in the lifecycle is it done?
4. **Who** – who's doing it?
5. **Process** – what is the process / method to design a software architecture ? (any tool support?).

6

From 1998 to 2020

1. **Semantics** – definition: what is it?
2. **Representation** – how does it look like?
3. **When** – in the lifecycle is it done?
4. **Who** – who's doing it?
5. **Process** – what is the process / method to design a software architecture ?

1. Semantics

- Hard to converge on a simple definition in the 1990s
- Paul Clements' collection of definitions
  - Structure, Component, Connectors
- Standard IEEE 1471:2000 on represetnation
  - "Software architecture refers to the fundamental structures of a software system and the discipline of creating such structures and systems. Each structure comprises software elements, relations among them, and properties of both elements and relations."

- Software architecture encompasses the set of significant decisions about the organization of a software system …

1995, with Booch, Reitman and Bittner
derived from Shaw and Garlan's definition

9

From the RUP (1998)

Software architecture encompasses the set of significant design decisions that shapes a software system, including the selection of the structural elements and their interfaces by which the system is composed; behavior as specified in collaboration among those elements; composition of these structural and behavioral elements into larger subsystems; and an architectural style that guides this organization.

Software architecture also involves functionality, usability, resilience, performance, reuse, comprehensibility, economic and technology constraints, tradeoffs and aesthetic concerns.

TL; DR

10

- *The architecture* is the set of *significant* design *decisions* that shape a software system, where *significant* is measured by cost of change.

Booch 2006

Rhymes with Ralph Johnson's, and Martin Fowler's views

Software architecture today

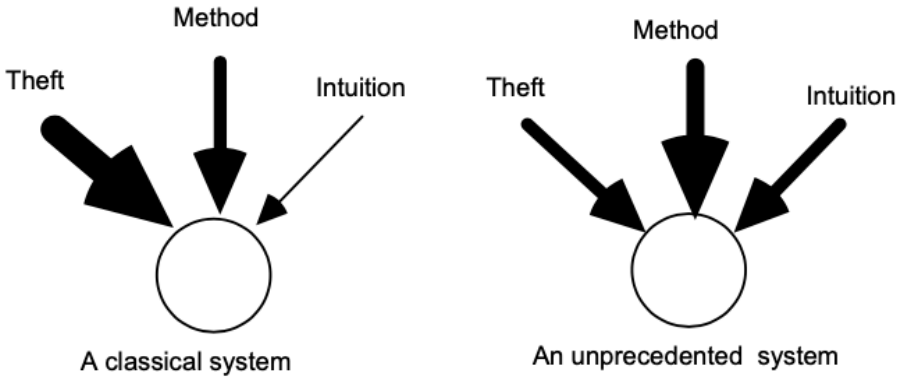
- Architecture as a common understanding of trajectory and boundaries.
  - Where is the project going
  - Shared mental model of "the design"
  - Conceptual integrity (handrails)

From 1998 to 2020

1. **Semantics** – definition: what is it?
2. **Representation** – how does it look like?
3. **When** – in the lifecycle is it done?
4. **Who** – who's doing it?
5. **Process** – what is the process / method to design a software architecture ?

5. Process – the recipe

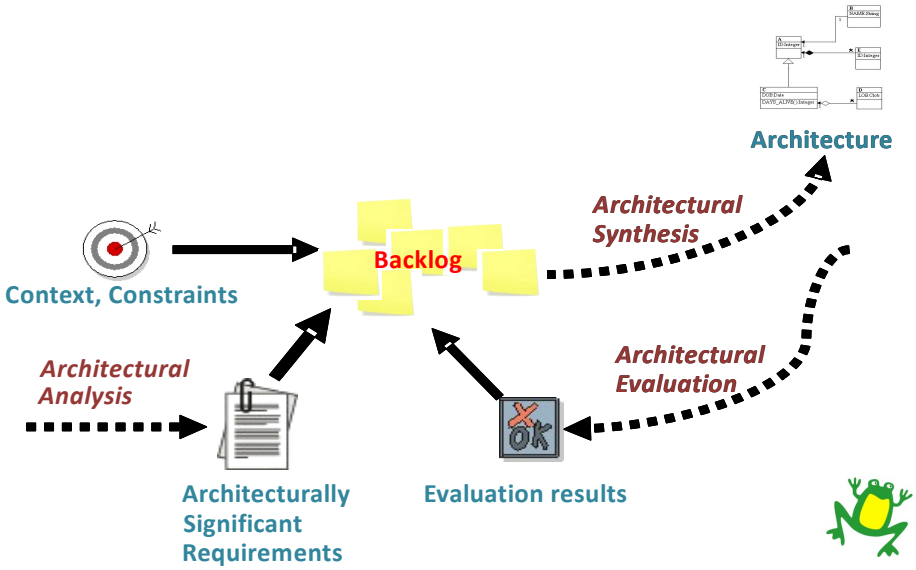
- "Mommy, where do software architecture come from?"
  – ICSE 1995 workshop on architecture of software system
- Theft, method, intuition



ICSE 1995

15

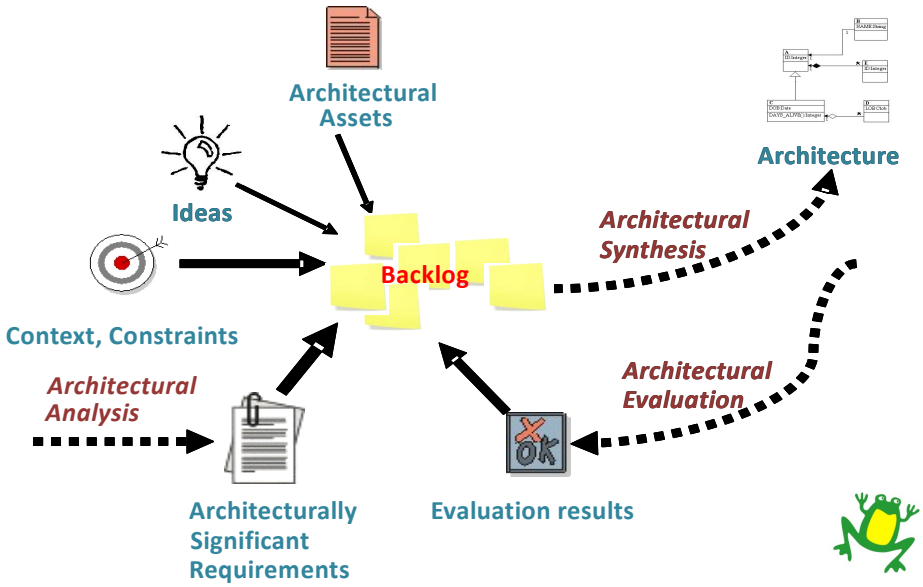
Architectural Design Method



Source: Hofmeister, Kruchten, et al., 2005, 2007

16

Architectural Design



Source: Hofmeister, Kruchten, et al., 2005, 2007

17

17

Useful input: elements of design

- Catalogue of Styles (Shaw and Garlan)
- Architectural Patterns (Stal, Buschmann & co)
- Architectural mechanisms (in RUP), also called Architectural Tactics at the SEI (Bass, Kazman, Clements & co)

- ....

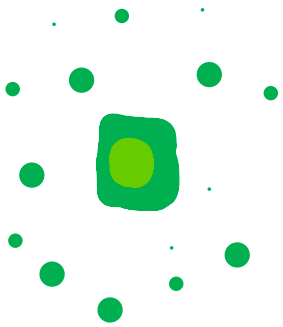
18

Today's situation

- Many accessible sets of architectural solutions
- Covering a wide range of problems or domains
- Documented
- Tested
- Many are open-source
- Some are developed or used by large companies
- Large active communities
- Reduced risks

Architectural nugget

- Concrete implementation of one or a small set of architectural patterns
- Around a programming language
- Stacks, ecosystems, solution framework, …



Examples

- LAMP stack

- MEAN stack
  - MongoDB, Express, AngularJS, NodeJS

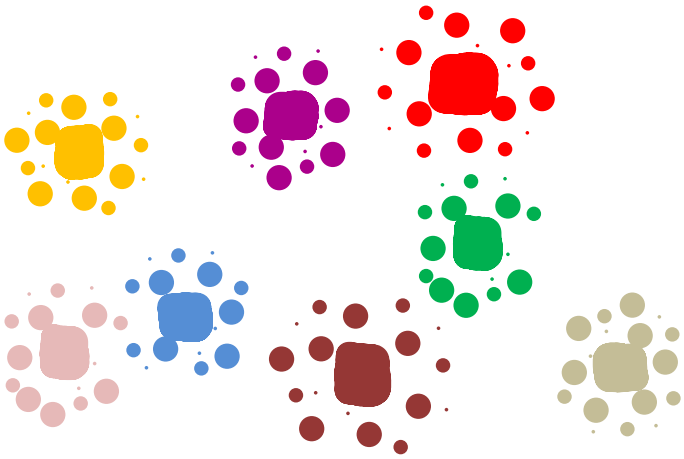
- Ruby on Rails ecosystem

- REACT Native

And dozens more

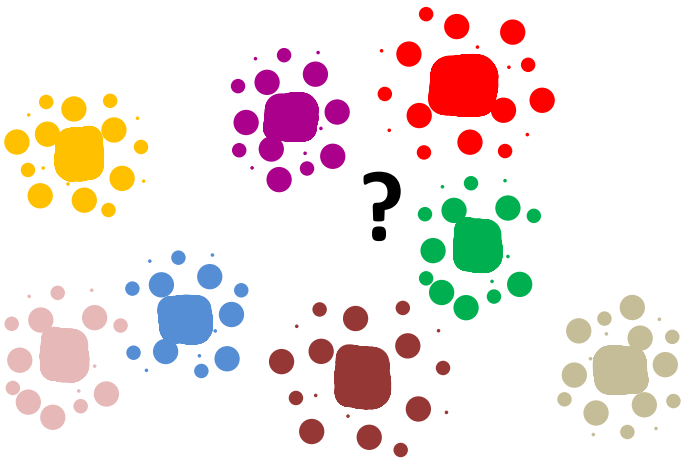
21

Today's landscape



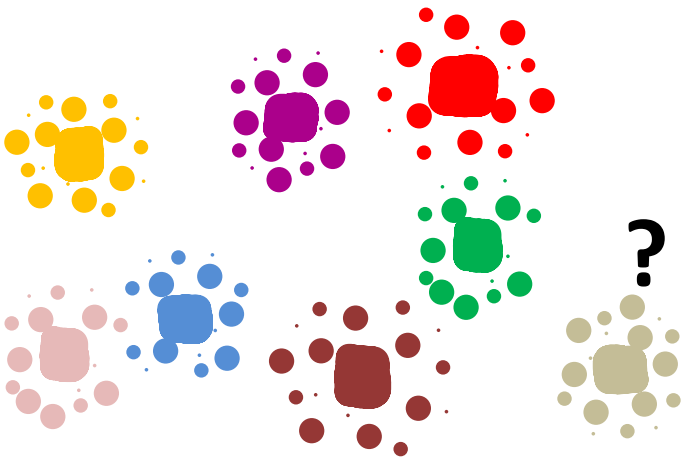
22

Pick one



23

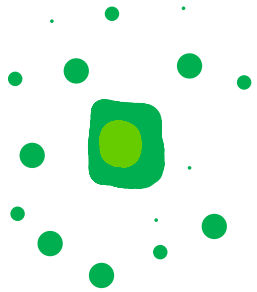
Less suitable choices



24

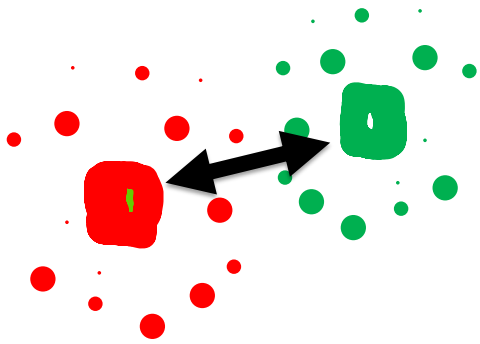
Our architectural nuggets…

- Implicitly impose an architectural style
- Coherence, consistency
- Interfaces, API



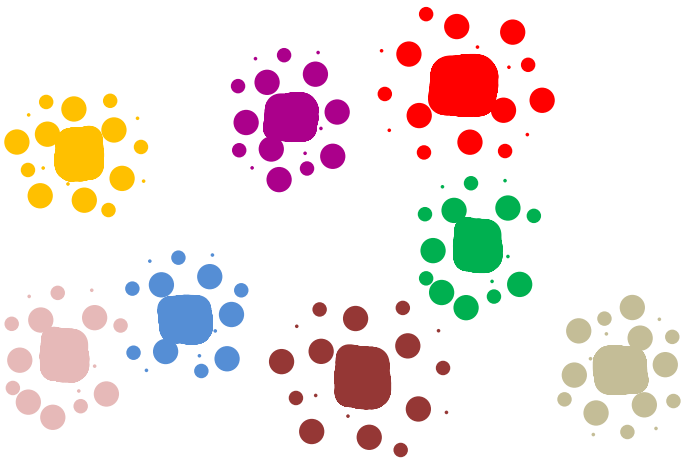
25

Impedance mismatch

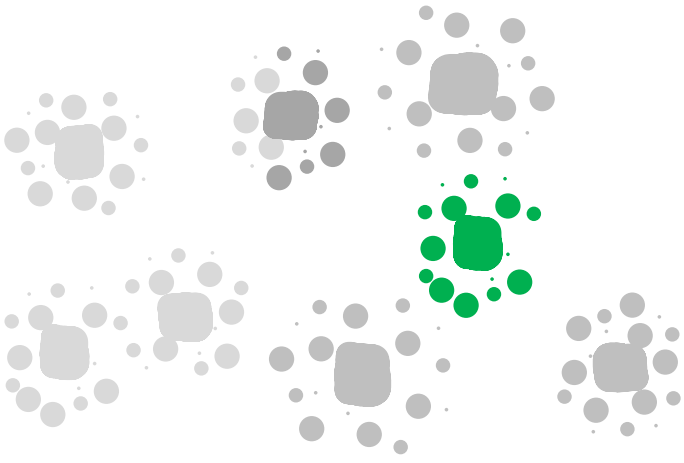


26

Maybe no good choices



27



28

Landscape has changed a lot

- Pick a general trajectory
  - Comes with a set of ready-made decisions
  - Experiment
  - Adapt
- Most of the remaining architectural work is about resolving impedance mismatch

29

Architecting today

- New issue or concern X => architectural backlog
- How are the others doing in our ecosystem?
  - Let us do the same they did for X
- We need something new or different?
  - How do we make to work? (impedance mismatch)
  - Risk? Let us experiment now

30

Architecting today

- Mostly theft (= select, copy, adapt)
- Very little intuition (= invention)

In general

- Not much need for rigorous step by step method
- On the down side:
  - *Trapped in a solution path*
  - *Architectural debt*

From 1998 to 2020

1. **Semantics** – definition: what is it?
2. **Representation** – how does it look like?
3. **When** – in the lifecycle is it done?
4. **Who** – who's doing it?
5. **Process** – what is the process / method to design a software architecture ?

4. Who - the software architects

- Software architect
  - Job title, function, hierarchy,
  - badge of honour for services rendered, ivory tower
- Educating architects
- Certification
- A few large places have succeeded doing so (e.g., Siemens)

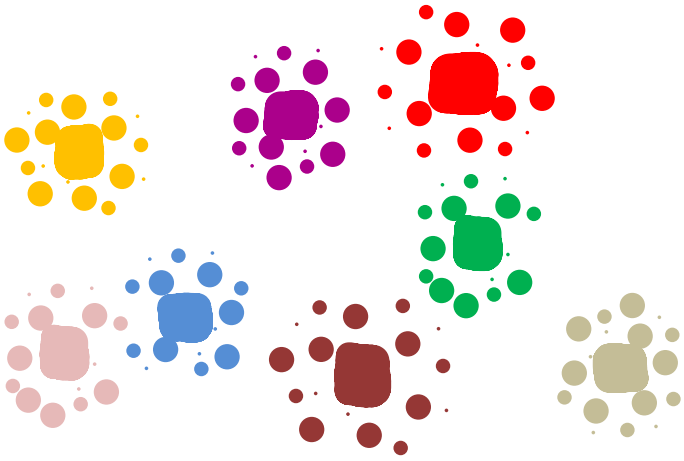
33

4. Who - the software architects

- Issue with the title?
- Architecture is design
- Experience is key
  - Little material available in the 1980s – 90s
  - Lots of visible architecture today
    - Open source
    - Major players (Google, Amazon, Netflix, Spotify, …)
    - ecosystems

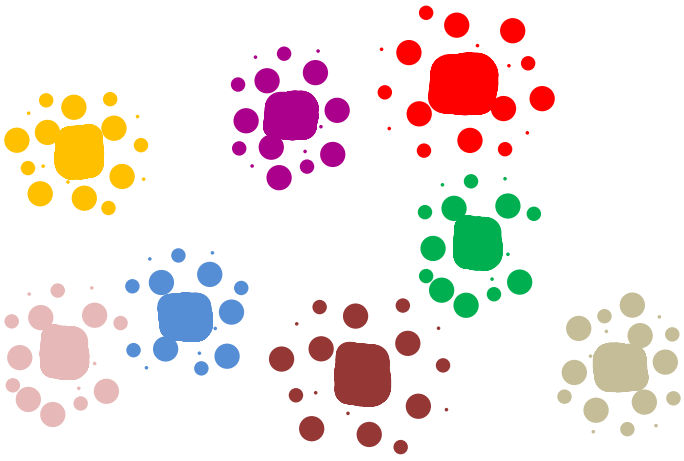
34

Experience



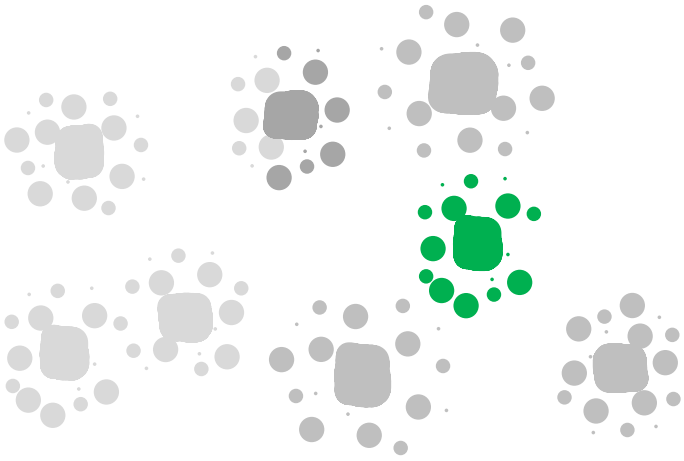
35

Breadth of Experience



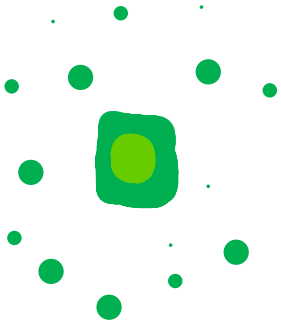
36

Breadth of Experience



37

Depth of Expertise



38

- Martin Fowler, Who needs an architect (2003):
  - Architectus reloadus => more breadth)
  - Architectus aryzus => more depth

- Architecture Owner

- How do you becomes an architect?
  - Course, books? Meh…
  - Practice, exposure to a wide array of cases

But not to forget:
- Decision making
- Cognitive biases

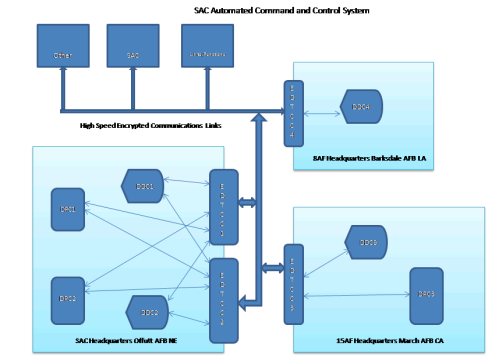
- Leadership
- Communication

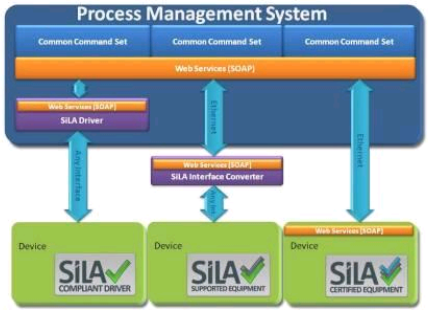
From 1998 to 2020

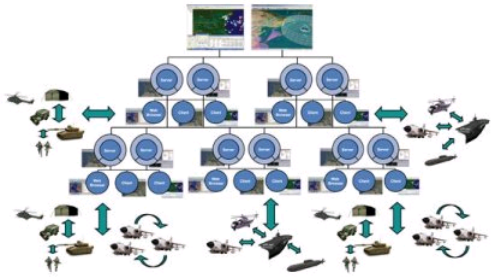
1. **Semantics** – definition: what is it?
2. **Representation** – how does it look like?
3. **When** – in the lifecycle is it done?
4. **Who** – who's doing it?
5. **Process** – what is the process / method to design a software architecture ?

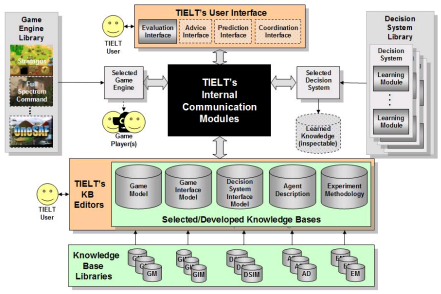
2. Representation

- 1990's
  - Boxes and arrows
  - Icons
  - Colors
  - *PowerPoint*









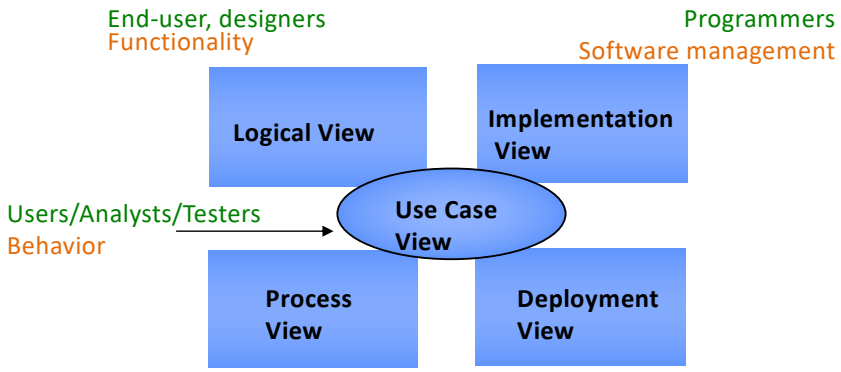
43

Views (ca. 1990)

- An idea that was ripe
  - Philips BAPO/CAFR
  - Siemens S4V
  - Rational's 4+1 views
- Took root for a while:
  - Eoin Woods & Nick Rozanski book
  - IEEE 1471:2000, ISO 42010:2007

44

The 4+1 view model of architecture



Kruchten 1995

45

Views in 2020

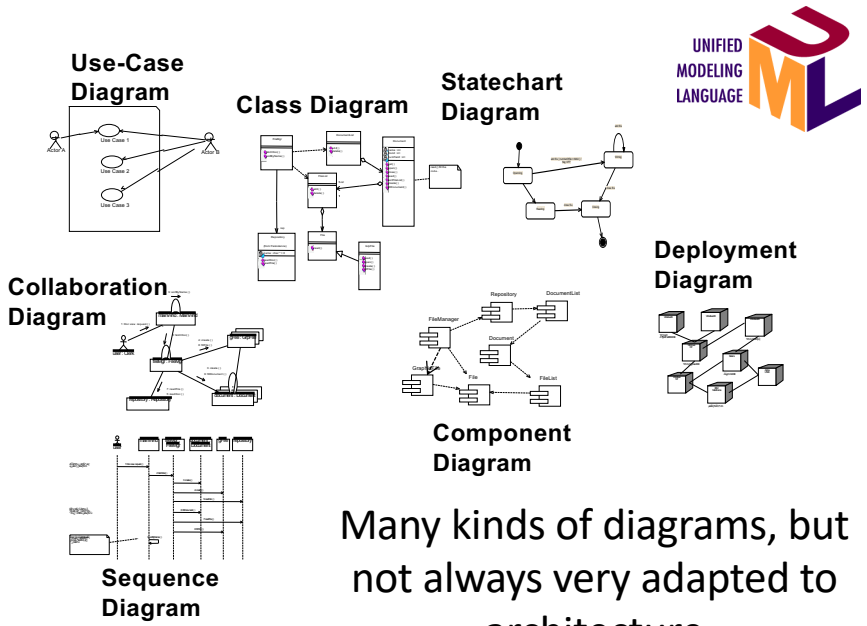
- Still a useful concept in 2020 (?)
- Not much evidence of actual application
  - Simon Brown's C4Model …?
- Visualization is still important
  - Ruth Malan & Dana B.
- Tool: back to PowerPoint

46

Small detour: ADL

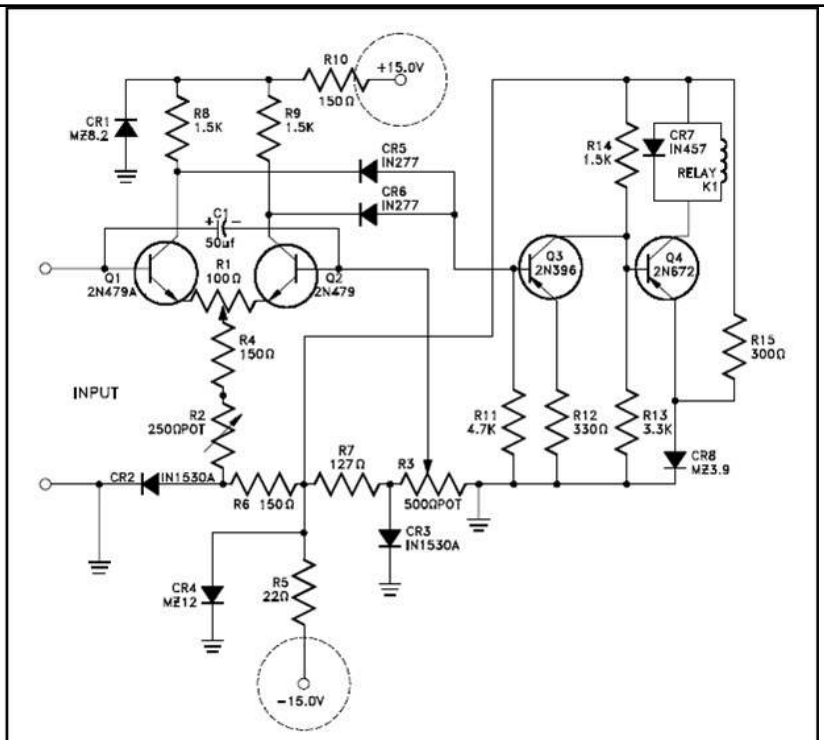
- Architecture Description Languages
- Many created (> 100)
- A handful more noticeable
- Almost no traction in industry
- UML 2.0 as an ADL? (2005)

47



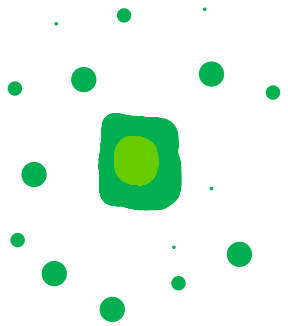
Many kinds of diagrams, but not always very adapted to architecture

48



Representation today

- Minimal documentation attached to pieces of an architectural nugget. No UML. No views.



From 1998 to 2020

1. **Semantics** – definition: what is it?
2. **Representation** – how does it look like?
3. **When** – in the lifecycle is it done?
4. **Who** – who's doing it?
5. **Process** – what is the process / method to design a software architecture ? (any tool support?).

51

3. When does it happens

- 1990: waterfall lifecycle still prevalent
- 200: Start of the never ending debate
  Big Up-front Design *versus* Gradual emergence
- "Big design up front is dumb. Doing no design up front is even dumber."

  Dave Thomas, via Simon Brown

52

Agile Manifesto - principle #11 (2000)

"The best architectures, requirements, and designs emerge from self-organizing teams."

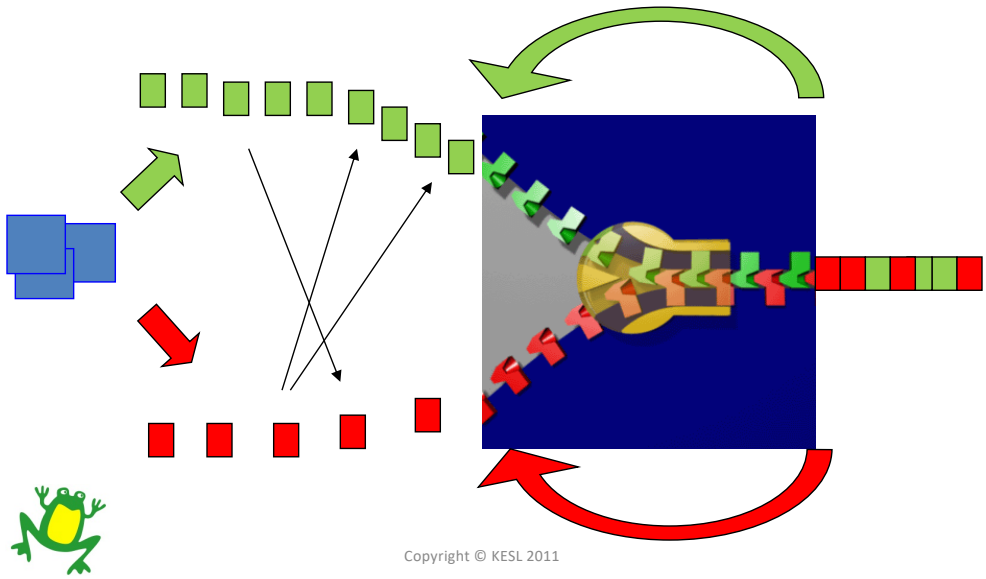
Really?

Refactoring

You need an "Architecture owner" to drive the emergence

53

Zipper model: Weaving functional and architectural bits



54

Actually, today….

- Once you've picked an architectural nugget, *emergence* is pretty fast:
  - There are ready-made, validated solutions to most issues or architectural concerns
  - Narrow range of alternatives: stick with the family
- Architecture is slowly incorporated in more mature agile approaches:
  - SAFE, or LESS, for example

Recap: Software Architecture in 2020

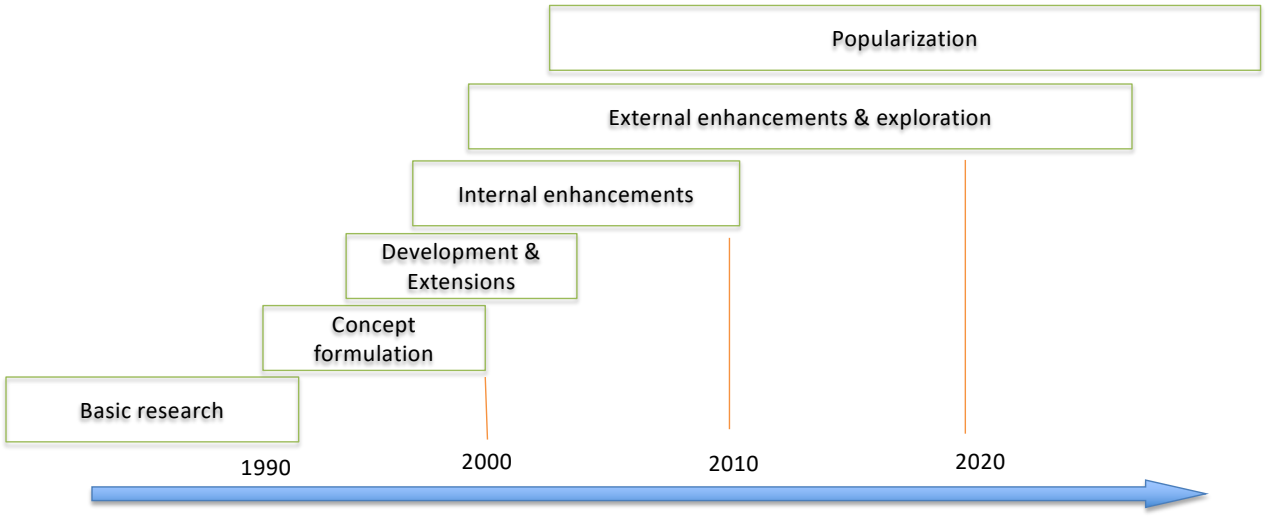
1. **Semantics** – roughly understood
2. **Representation** – Text and code, boxes and arrows, some views and UML in rare cases
3. **When** – Iteratively, early in the life-cycle; occasional refactoring when no way forward
4. **Who** – a software designer with expertise and/or authority
5. **Process** – Steal, adapt, experiment, steal some more, adapt. Manage impedance mismatches.

A Mature Discipline?

1. Basic research
2. Concept formulation
3. Development and extension
4. Internal enhancements and exploration
5. External enhancements and exploration
6. Popularization

Redwine Riddle 1985
Shaw 2001
Shaw Clements 2006
Clements Shaw 2009

Mature Discipline?



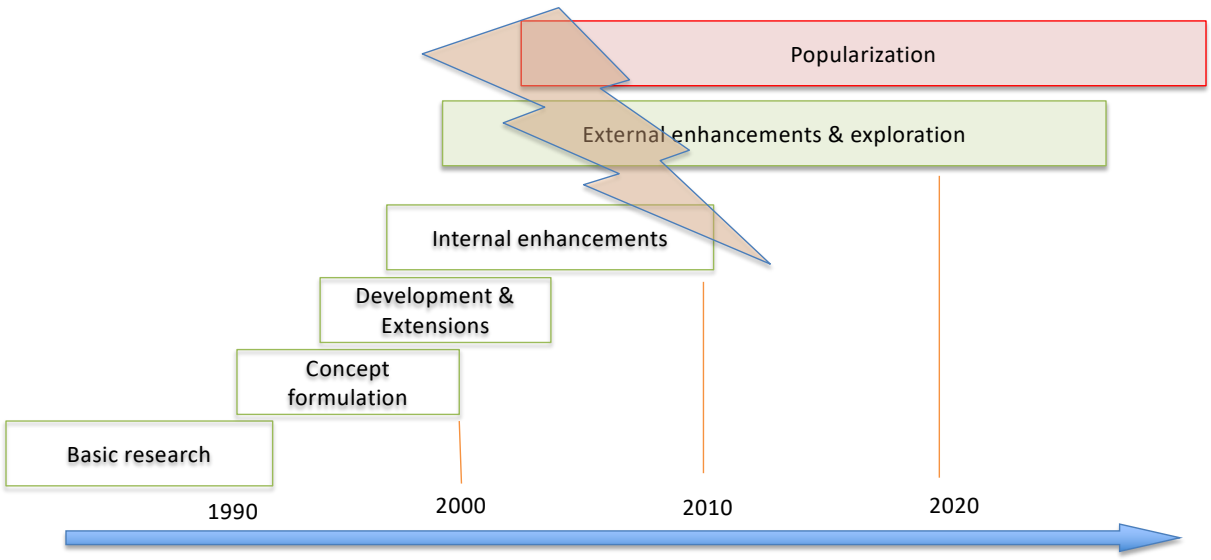
A mature discipline?

In 2020:

- Body of knowledge (books, etc)
- The landscape has changed
- Less general conferences, more specialized ones
  - Saturn, O'Reilly, CompArch + WICSA
  - Architecture ***and*** XYZ
    - where XYZ = DevOps, MicroServices, BlockChain, Tech Debt, HealthCare, Security, IoT, Social aspect, Industry 4.0, …

59

Mind the gap ?



60

A mature discipline?

External enhancements and exploration

- Driven by
  - Open source
  - New technologies
  - Speed
- Left behind much of the earlier progress
- Discontinuity, Gap
  - Left behind: definitions, standards, methods, representation

A mature discipline?

Popularization

- We have not done such a great job at popularization (hence the gap)
- How to guides at O'Reilly
  - The first book ever on software architecture was just published in 2020 (no kidding).
- Diversity of approaches (which is good)
  - More than one definition
  - More than one representation
  - More than one method
- Some standards

"Everything has been said before, but since nobody listens, everything must be said again." *André Gide 1891*

Where do we go from here?

- *External enhancement and exploration*
- *Popularization*

- On Software architecture:
  - Few key individuals
  - Develop their own ideas
  - Become star speakers at conferences
  - Write blogs and books
- On architectural nuggets:
  - Few key companies or groups of individuals
  - Develop their architectural solution
  - Create a community or ecosystem around it
  - Speak at more specialized conferences
  - And How-to books are published at O'Reilly

63

Where do we go from here?

- *Internal enhancements and exploration*

- Research on software architecture
  - Validate or invalidate some of the new ideas
  - Software Architecture *& something*
    - Technology, domain, concern

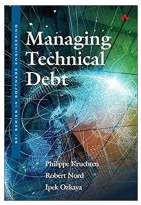
  - *"Under the lamppost"*
  - *Constrained by the publishing game and the size of PhD chunk*

64

Topics needing more research

- Economic aspects
  - Value & cost
  - Technical debt (as an angle on this)
- Social aspects
  - Role, leadership, communication
  - Open-source development dynamics
- Decision making process, and cognitive biases
- Representation & Tool support
- Tactics

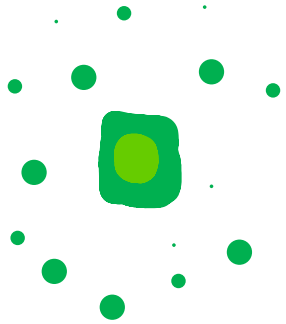
hint, hint … =>



65

"Mommy, where do the architectural nuggets come from?"

- Who, How, Why,….



66

Thank you….

… Linda, James, the SEI, and all of our colleagues around the world

67

People to thank

M. Shaw, D. Garlan, D. Perry, A. Wolf, G. Booch, M. Fowler, I. Gorton, R. Malan & D. Bredemeyer, E. Rechtin & M. Maier, R. Kazman, P. Clements, L. Bass, C. Hofmeister, H. Obbink, H. van Vliet, P. Lago, J. Bosch, F. Buschmann, M. Stal, C. Larman, A. Tang, E. Woods, E. Poort, G. Hohpe, R. Hilliard, U. Olson, M. Klein, J. Klein, L. Northrop, R. Martin, M. Feathers, J. Coplien, C. Szyperski, R. Nord, I. Ozkaya, *The Siemens & Philips & ABB folks*, W. Kozaczynski, R. Wirfs-Brock, D. Sotirovski, D. de Koning, B. Hamilton, J. Barnes, N. Kruchten, *and many more*

68