Architecting as Decision Making

Hans van Vliet
VU University Amsterdam

My personal history

1967	computer operator, programmer
1973-1978	MSc Mathematics/CS
1979	PhD, ALGOL 68
1986	Professor Software Engineering, VU University
1993	Software Engineering textbook (2000, 2008)
2008	Journal of Systems and Software (EiC)
1996-	Research Software Architecture (ALMA, GRIFFIN, Stephenson), GSD, agile, social aspects


Where did it all start?

- From SA = “components + connectors”
- to SA = “set of design decisions” (Bosch 2005)
- to SA = both (solution + why of solution)
- → Capture design decisions/rationale
- → Architectural knowledge & its management

Capture design decisions/rationale

- Explicitly **document** design decisions (WICSA 2002)
- Janet Burge ~2002
- Document design rationale (IBIS–1979, gIBIS-1987)
- Design space analysis (QOC, 1991), especially in HCI

Architecture knowledge

- Encapsulated in patterns
- Codified in dynamic architectures (usually graph structures)
- In detailed requirements (co-development of requirements and architecture -- twin peaks)
- In design decisions

What this brought us

- Many tools to capture design decisions/rationale
- Architecture approaches that emphasize design decisions, such as RCDA

Key Principles of RCDA, Risk and Cost-Driven Architecture

Cost and Risks drive architecture
- highest impact on *cost and risks* of the system and its delivery
- architect should be an expert on costing and risk mitigation

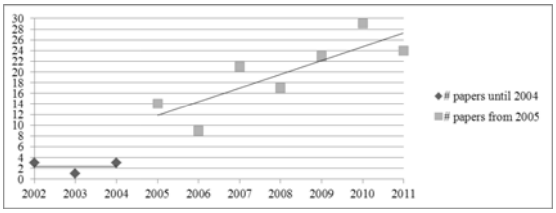
Architecture should be minimal
- to keep overview of the whole system
- Solution Architect should limit to decisions with critical impact
- leave a maximum of design space for developers

Architecture as both Blueprint and Decisions
- decisions leading to architecture and the underlying rationale are essential

Solution Architect as Decision Maker
- critical architectural decisions are made by *one person* with overview of whole system
- requires authority and subject matter skills and knowledge

Courtesy Eltjo Poort, CGI, 2013

Software architecture decision papers



From: Dan Tofan et al, Past and Future of Software Architectural Decisions - A Systematic Mapping Study, IST, 2014

So, the architect makes decisions

- Rational? Irrational?
- Is she possibly biased?
- Decisions about what: solution, or problem?
- How important is the first decision?

Can we make rational decisions?

- Absolute rationality
  - Purely logical chain of events and consequences
  - Time-consuming
- Bounded rationality (Herbert Simon)
  - Our capabilities are limited -> heuristics, rules of thumb, "this works because it worked last time"
  - Time-efficient
- Social/cultural rationality
  - Our limits necessitate interdependence ("two know more than one")
  - Differences give new perspectives and solutions

Software design thinking styles

an experiment to find out how students and professionals reason about design situations

An example scenario

- A new system is to be created to simplify the government election process. There are 10 million eligible voters in the country. Each voter would be issued a smart-card. A smart-card is authenticated by password and a finger-print. Voters can vote through the Internet. It is expected that many voters would vote online. The number of voting booths and the number of employees required for vote counting can be reduced significantly.
- ***Conclusion***: The savings of an election would be significant.

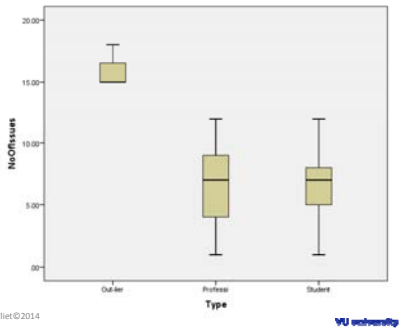
Issues identified by participants

1. Costs - Who is to pay for this device and how much does this device cost?
2. What is the finger print registration process and system?
3. Some people may opt not to use technologies, e.g. aged people.
4. What is the density of population and the location of booths?
5. What is the cost and ROI?
6. Number of people having online access.
7. The cost of educating voters.
8. This system requires a finger-print reader.
9. The system requires a finger print DB to be set up for 10 million people.
10. Government is able to obtain finger prints and backup policies.
11. Is there any privacy issue with the finger print registration?
12. Do we know that people would be willing to vote on-line? How many amongst the 10 million people would be required to make a saving?
13. The security risks, such as in collecting and maintaining personal data and finger prints; and detection of fraud or hacking activities.

Findings

- Students (32) are not different from professionals (29):
  - They find the same number of issues
  - They do not make different judgments
  - Both find few reasons in comparison to all the reasons that could be found
- BUT: some professionals behave VERY different from all other professionals and all students

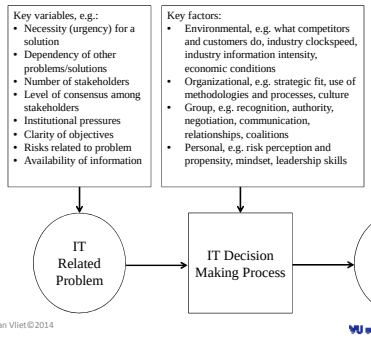
Issues by outlier professionals vs the rest



Analysis of behavior of outlier professionals

- They use less analogy
- They provide (many) more reasons
- Outliers use analytical/rational thinking
- The rest uses intuition: they follow the Law of Least Effort, use minimal cognitive load whenever they can

Decision making in Enterprise Architecture



Design as a "wicked" problem

- There is no definite formulation
- There is no stopping rule
- Solutions are not simply true or false
- Every wicked problem is a symptom of another problem

Design problems are dilemmas

- Cannot be stated per se
  - Depends on context/environment
- Cannot be solved with a definite answer
  - Multi-dimensional (stakeholders, concerns, constraints, …)
- Are complex, full of implicit, overstated demands
  - E.g. of customers
- Any solution generates (often unknown) "waves of consequences"
- Calls for creativity and ingenuity

Moran & Carroll, Design Rationale, 1996

So, the architect makes decisions

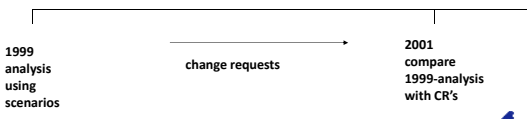
- Rational? Irrational?
- **Is she possibly biased?**
- Decisions about what: solution, or problem?
- How important is the first decision?

Bounded rationality ⇒ role of context

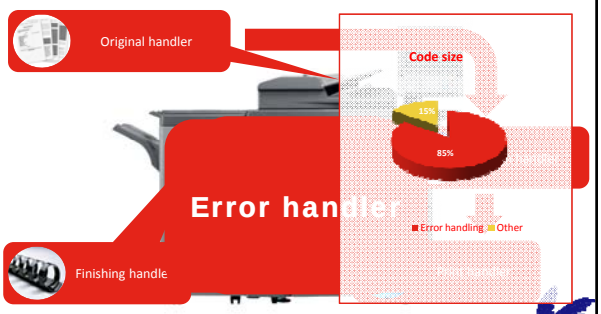
- Decisions are made in a context
- Selecting a context is a pre-decision act, mostly done unconsciously, self-steering, based on experience (e.g. automatic adjustments when driving in a busy street)
- Anchoring, … and other biases

How well can we predict changes?

- PhD research Nico Lassing, 1997-2001
- Theme: Architecture-Level Modifiability Analysis (ALMA)



Software architecture of a copier



Courtesy Robert Deckers

From requirements to design creativity

- Group A: "your task is to develop one or more design concepts for … an analyst has developed … the following *requirements specification*: …"
- Group B: "your task is to develop one or more design concepts for … an analyst has developed … the following *list of ideas*: …"

Requirements fixation

- Group A produced significantly less original designs
- → framing desiderata as "requirements" causes fixation: designers get preoccupied with satisfycing the requirements, rather than creativity

Source: Mohanani, "Requirements Fixation", ICSE 2014

Cognitive Bias

- The notion of cognitive biases was introduced by Daniel Kahneman and Amos Tversky in 1972.
- A cognitive bias is a heuristic, a simple rule that simplifies processing
- … but may easily introduce errors
- Examples of cognitive bias in SA:
  - "I opt for a SOA architecture (, since that worked on my last assignment")

Types of cognitive Biases

- Statistical
  - Sample – It worked once, so …
- Memory
  - Recall – what happened recently is important
- Confidence
  - Confirmation – what confirms your ideas is more important
- Adjustment
  - Anchoring – adjust from initial position
- Presentation
  - Order – see first or last item as more important
- Situation
  - Habit – choose same solution as before

Courtesy Arnott: Cognitive biases and decision support systems development

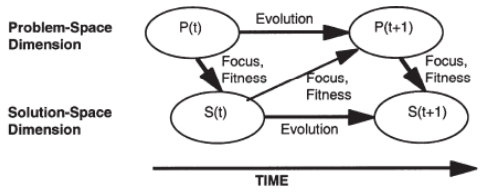
So, the architect makes decisions

- Rational? Irrational?
- Is she possibly biased?
- **Decisions about what: solution, or problem?**
- How important is the first decision?

Interaction between problem and solution

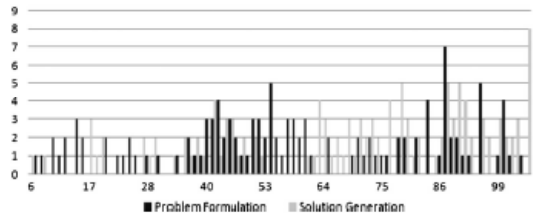
- "your requirement is my decision"
- Decisions lead to new problems and requirements, which need further decisions …
- Twinpeaks workshop series

Problem-solution co-evolution



Dorst/Cross: Creativity in the design process: co-evolution of Problem-solution, Design Studies, 2001

Example co-evolution



Example co-evolution

- 6.27: "So it's like a drawing tool …" (DD)
- 7.10: "I don't know if they can set the speeds. They can set the density…" (Req)
- 7.44: "We need some kind of visualization of the map" (DD)
- 8.03: "I don't know if there''d be two modes: an editing mode and a simulation mode…" (Req)

So, the architect makes decisions

- Rational? Irrational?
- Is she possibly biased?
- Decisions about what: solution, or problem?
- **How important is the first decision?**

First decision, and its impact

- Example: traffic simulation problem
  - MVC: focus on data structure, modeling, representation
  - Drawing tool: which part of the simulation is on-screen, scrolling buttons

Architecture hospital system

Another possibility

So there's many issues

- Bounded rationality
- All sorts of biases
- Which problem are we trying to solve
- Role of first decisions

How to fight all of this?

- **Recognize/acknowledge** the role of irrationality in design
- Include **debiasing** steps, e.g. through a checklist
- Pay **explicit** attention to "problem options" and "solution options" in architecture design
- Pay attention to problem framing in architecture design (importance of **first** decisions)
- Teach students to recognize these aspects

Debiasing checklist for lightweight architecture reviews

- Are there reasons to suspect motivated errors, caused by self-interest of architects?
- Have architects fallen in love with their decisions?
- Where there dissenting opinions in the design team? (Groupthink)
- Is the situation overly influenced by salient analogies? (proper context chosen?)
- Have credible alternatives been considered?
- Where do the numbers come from? (anchoring bias)
- Are the architects overly attached to previous decisions?

Inspired by Kahneman, Harvard Business Review, 2011

Thanks to

- GRIFFIN project (Jan Bosch, Patricia Lago, Paris Avgeriou, Remco de Boer, Rik Farenhorst, Victor Clerc, Anton Jansen)
- Antony Tang

References

- **Enterprise Architects Should Follow the Money,** Martin van de Berg and Hans van Vliet, *IEEE Conference on Business Informatics*,pp 135-142, 2014
- **Design Strategy and Software Design Effectiveness,** Antony Tang and Hans van Vliet, *IEEE Software 29(1), pp 51-55, 2012*.
- **What makes Software Design Effective** : **Antony Tang and Aldeida Aleti and Janet Burge and Hans van Vliet**. *Design Studies 31(6), 2010, pp 614-640*
- **On the Similarity between Requirements and Architecture** : **Remco C. de Boer and Hans van Vliet**. *Journal of Systems and Software 82(3), 2009, pp 544-550*
- **Design Reasoning Improves Software Design Quality** : **Antony Tang and Minh H. Tran and Jun Han and Hans van Vliet**. *Proceedings 4th International Conference on Quality of Software Architectures (QoSA2008), Springer LNCS 5281, 2008, pp 28-42*