Decision-making Processes and Cognitive Biases in Designing Software Architectures

Akash Manjunath, Garching, 19.02.2018

Software Engineering betrieblicher Informationssysteme (sebis)
Fakultät für Informatik
Technische Universität München

wwwmatthes.in.tum.de

Agenda

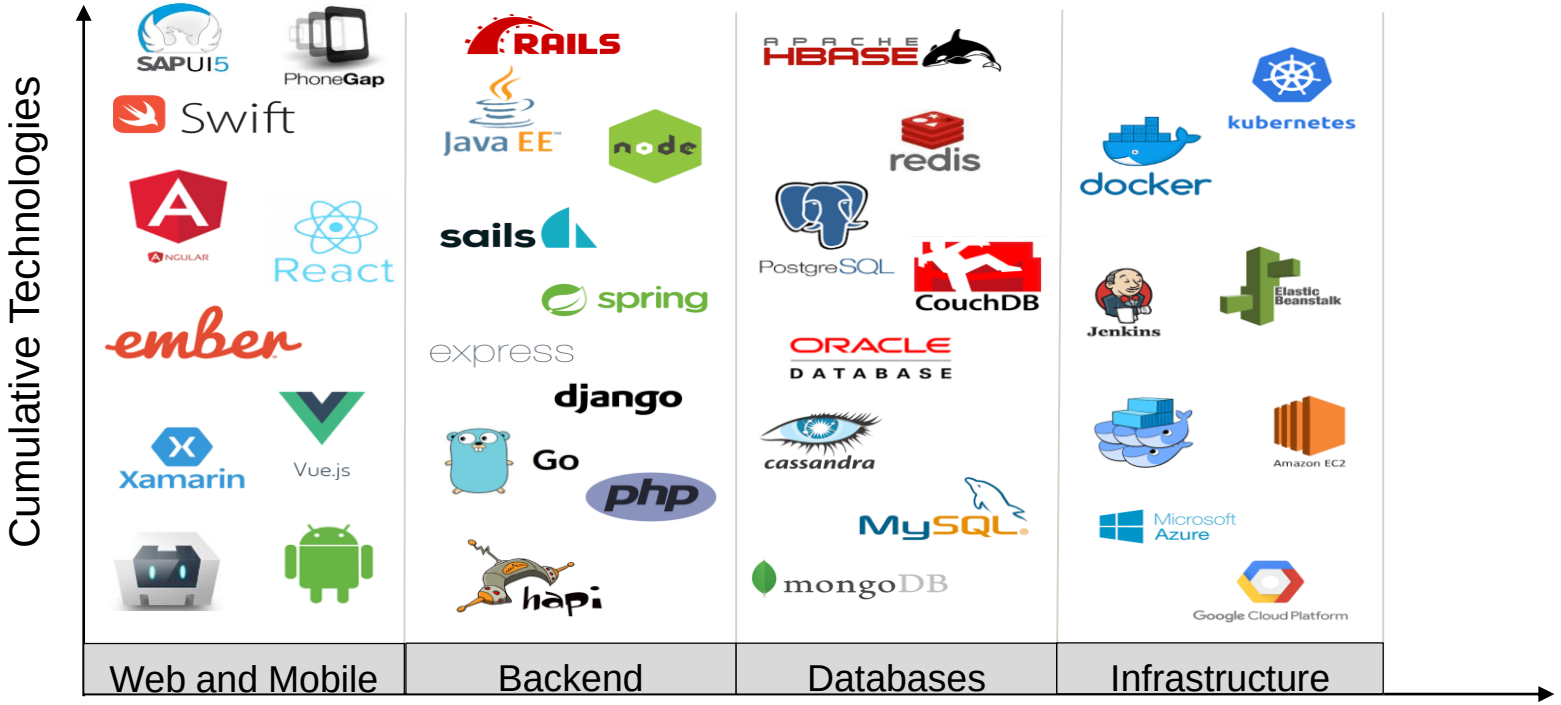
- Introduction
- Goals
- Research Questions
- Expert Reviews
- Research Implications

Introduction

- What is Software Architecture?
  - Blueprint about the structure of the software
  - Responsibility of software architects
    - Designing the architecture involves decision-making
      - Examples: Which design patterns to choose? Which combination of technologies to use? etc.

Challenge for Software Architects

- Too many technologies to mix and match
- Decision-making
  - complex, implicit and knowledge intensive
  - process is not well understood
- Heuristics such as past experience, familiarity, trends etc. are used for decision-making
- Decisions are biased due to the use of heuristics
- Results in *sub-optimal* or *satisficing* solutions



Goals

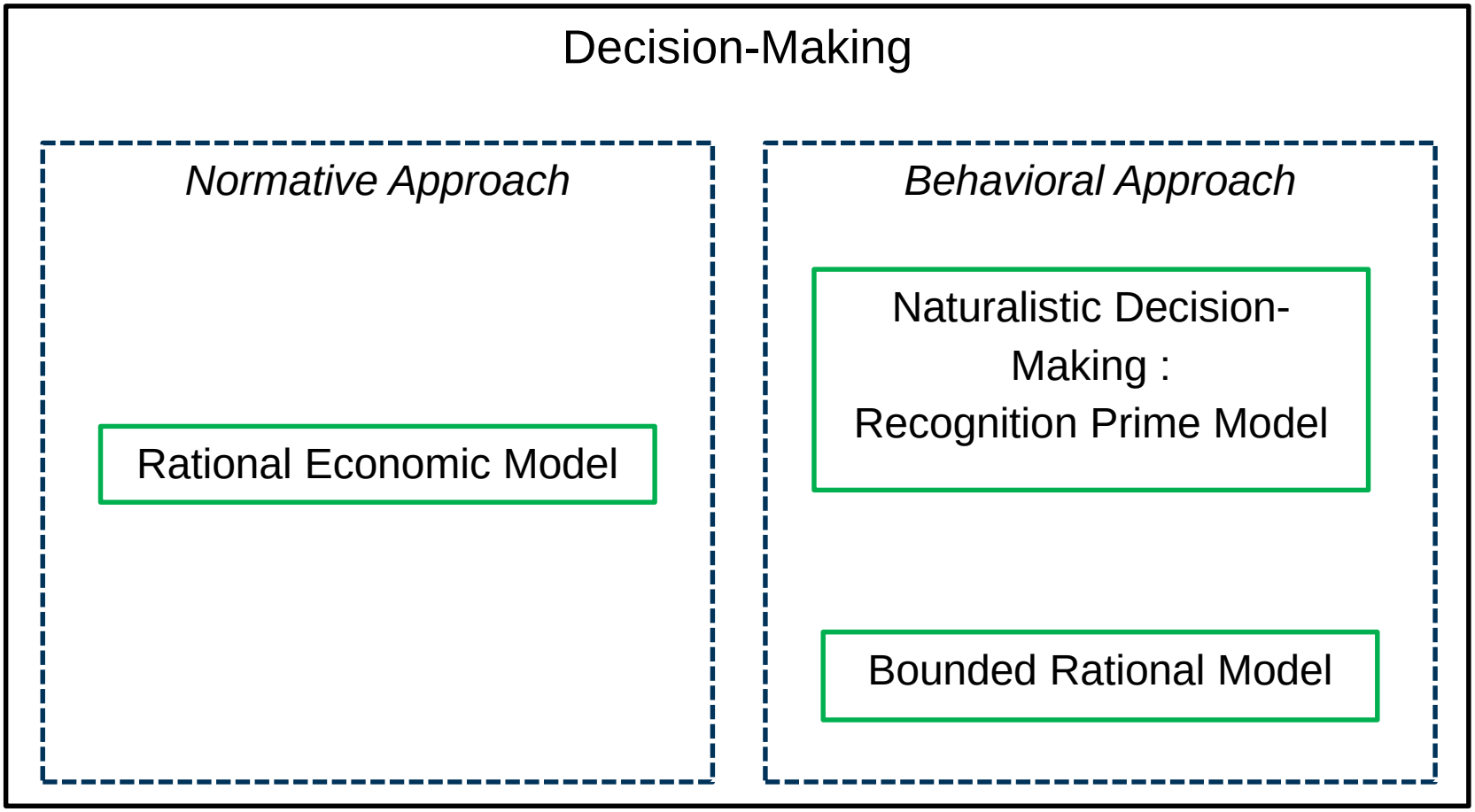
- From the context of designing software architectures
  - To formalize the decision-making process to make it explicit
  - To understand which cognitive biases influence software architects when designing architectures

Research Questions

1. **Which decision-making models are relevant in the context of making software architecture design decisions?**
2. What is the relationship between the decision-making models and the OODA loop?
3. Which cognitive biases influence software architects when designing architectures?

Decision-Making Models

- Several models of decision-making
- Focus on models relevant in the context of designing software architectures

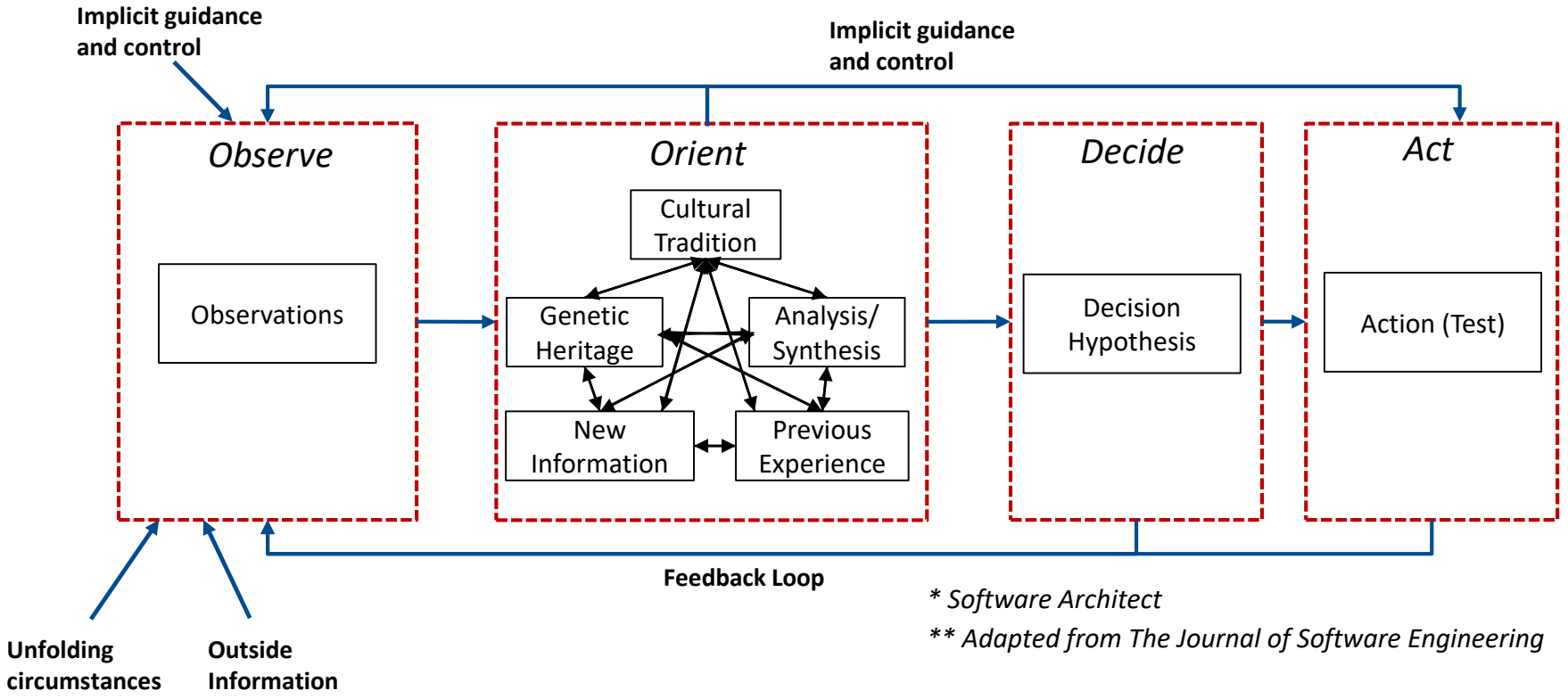


Research Questions

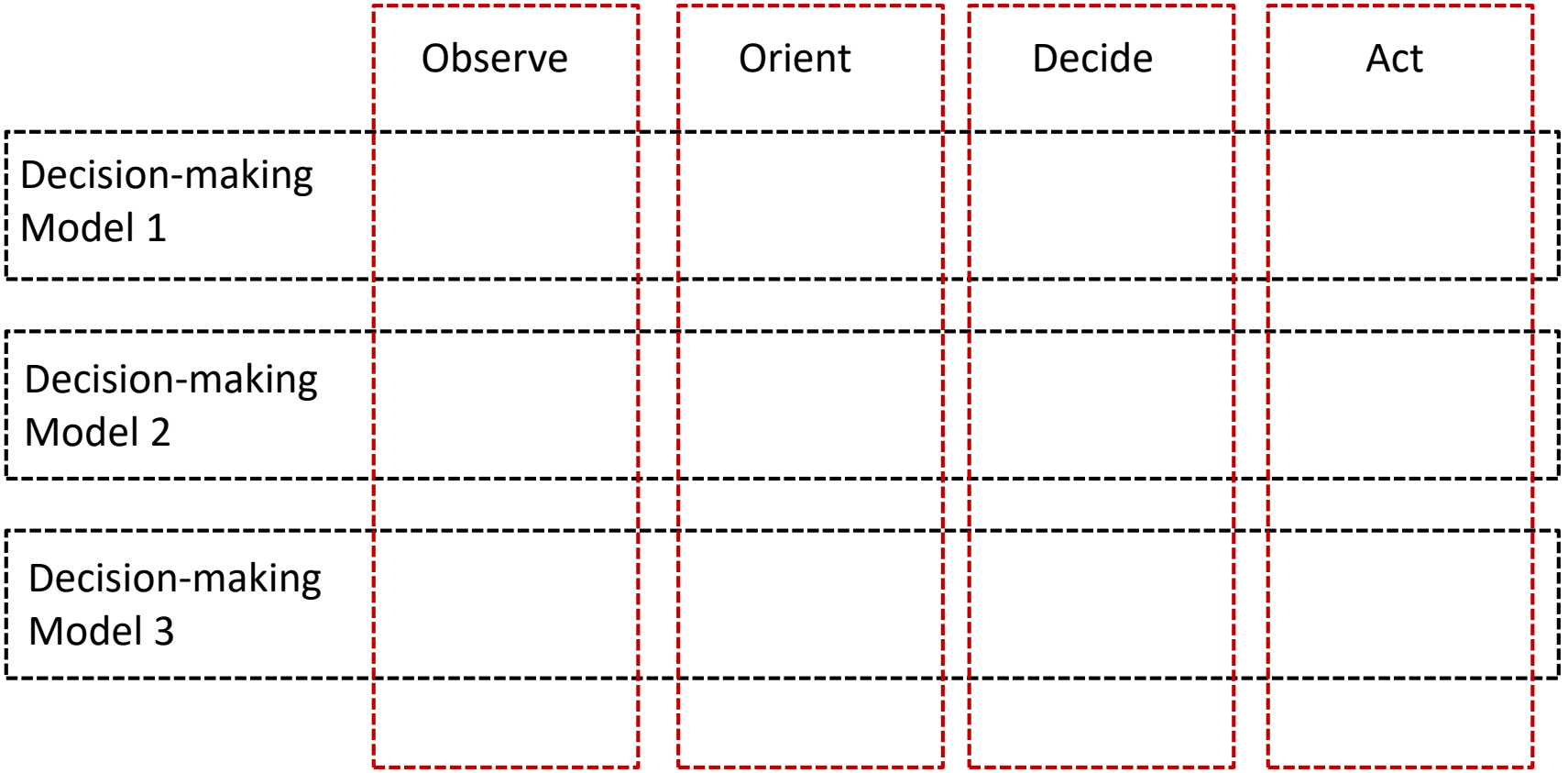
1. Which decision-making models are relevant in the context of making software architecture design decisions?
2. **How can we establish a relationship between the models with the OODA loop decision cycle?**
3. Which cognitive biases influence software architects when designing software architectures?

OODA Loop
Observe, Orient, Decide and Act

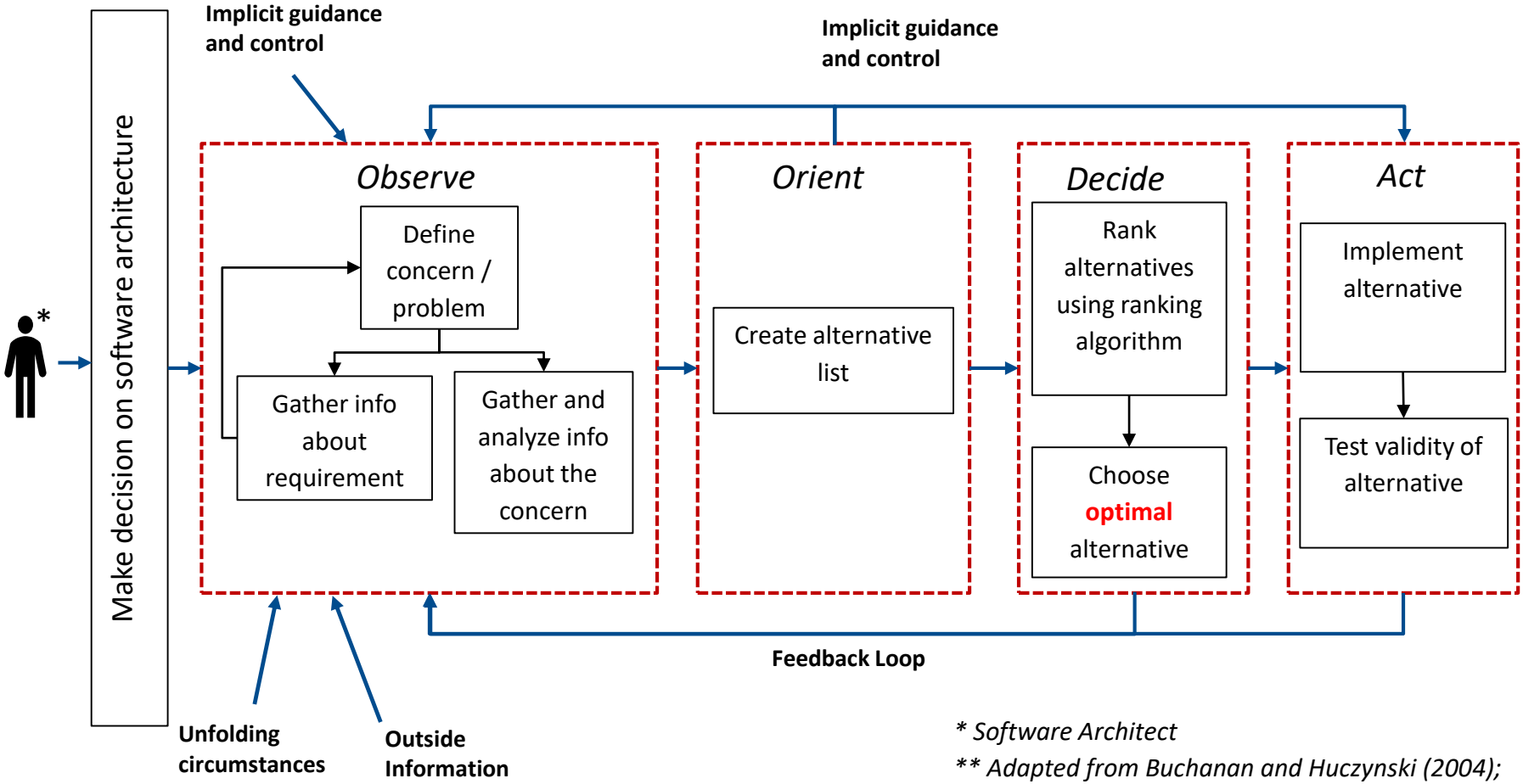
- Decision cycle proposed by John Boyd, a military strategist
  - Popular tool used by decision-makers in different fields of work
  - Much research conducted on OODA Loop and DMMs, but not inside the boundaries of designing software architectures



Matrix of DMMs and OODA Loop



Normative Model
Rational Economic Model **

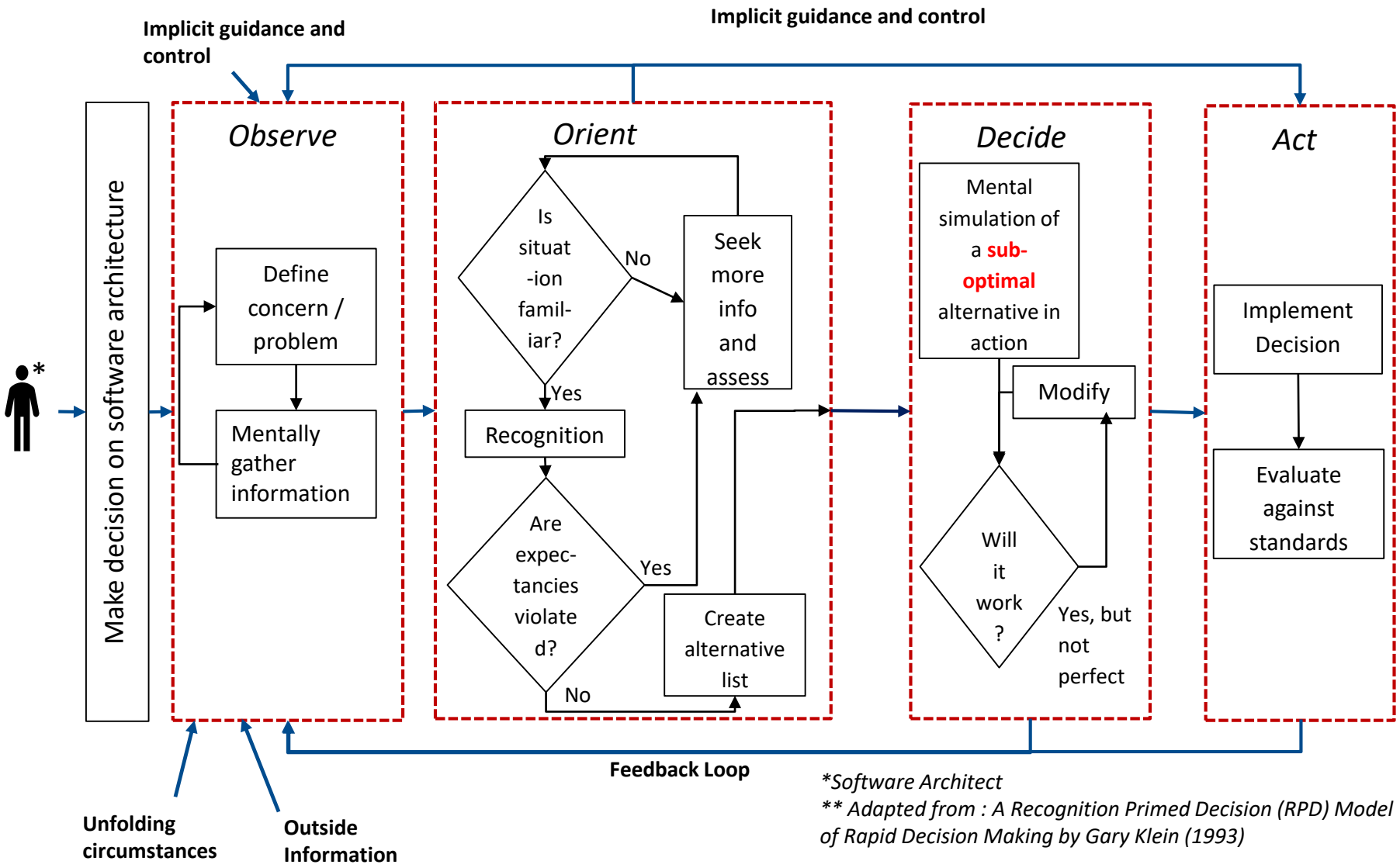


** Software Architect*

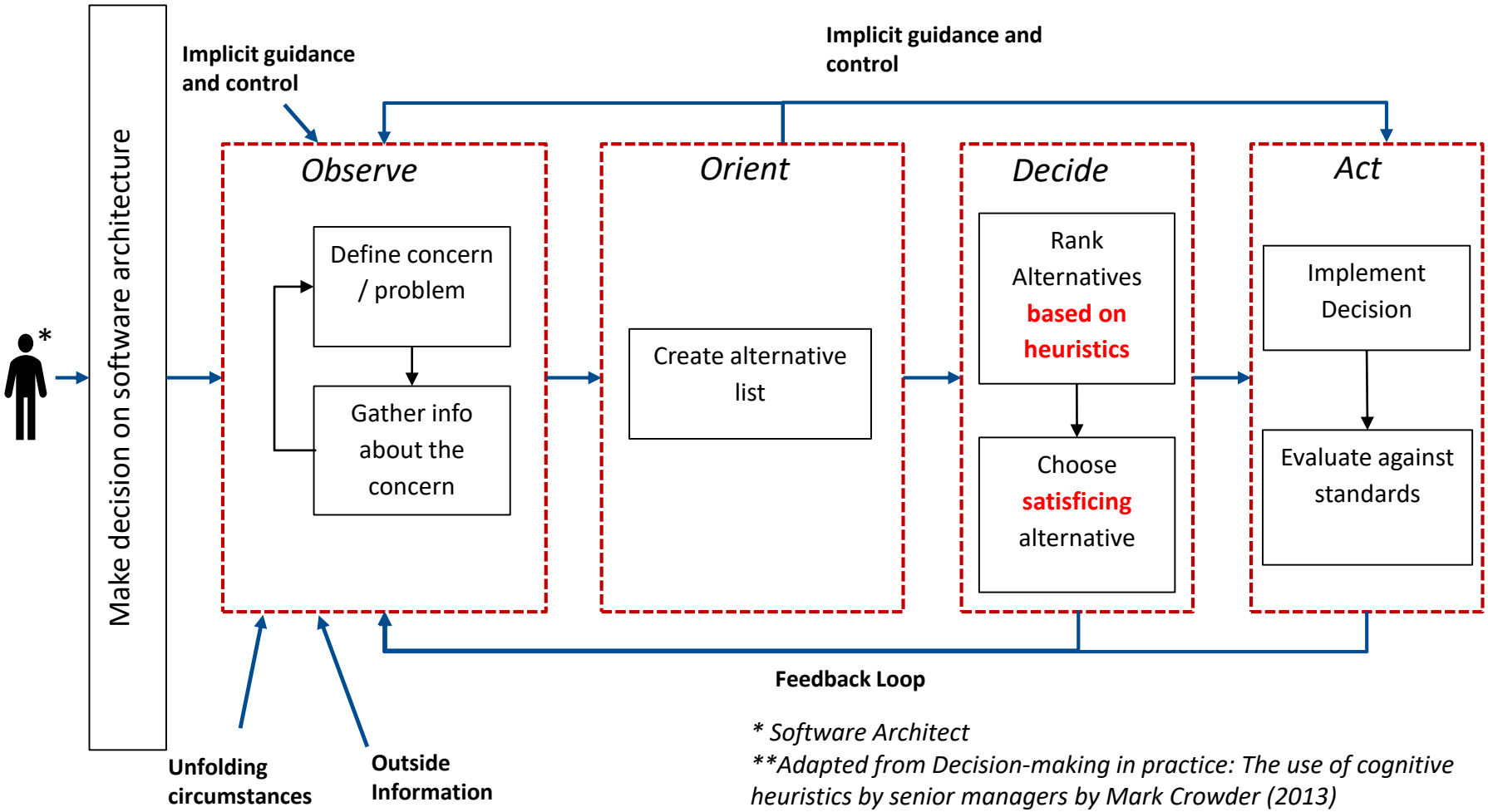
*** Adapted from Buchanan and Huczynski (2004); Drucker (2001); Miller Hickson and Wilson (2002).*

Behavioral Model

Naturalistic Decision Making – Recognition Prime Model



Behavioral Model
Bounded Rational **



** Software Architect*
***Adapted from Decision-making in practice: The use of cognitive heuristics by senior managers by Mark Crowder (2013)*

Research Questions

1. Which decision-making models are relevant in the context of making software architecture design decisions?
2. How can we establish a relationship between the models with the OODA loop decision cycle?
3. **Which cognitive biases influence software architects when designing software architectures?**

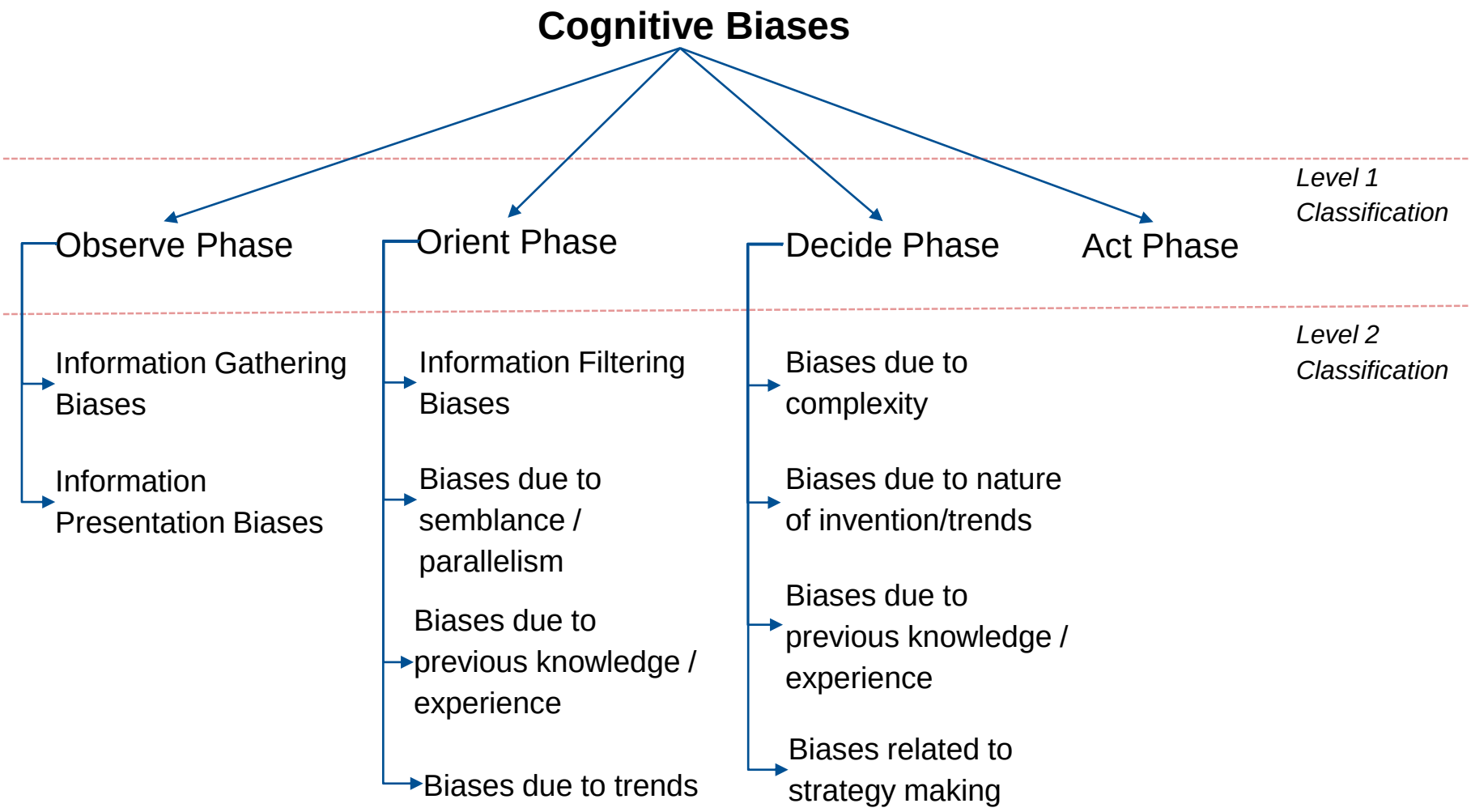
Cognitive Biases

- What are cognitive biases?
  - Systematic deviation from rationality in judgement
  - Due to limitations in human cognitive capacity
  - Impacts decisions and judgements
- Over **200** types of cognitive biases
- Not all of them are relevant for designing software architecture

33 cognitive biases recognized as relevant from the context of making architectural design decisions

Cognitive Biases Classification (1/2)

Two-level classification



Cognitive Biases Classification (2/2)

Cognitive Biases			
Observe Phase	**Orient Phase**	**Decide Phase**	**Act Phase**
Information Gathering Biases	***Information Filtering Bias***	***Biases related to Complexity***	Misinformation Effect
Completeness Bias	Base Rate Fallacy	Attenuation	Post-purchase Rationalization
Confirmation Bias	***Biases related to Semblance***	Hard-easy Effect	
Information Bias	Similarity Bias	Planning Fallacy	
Levels-of-processing Effect	***Biases related to Previous Knowledge / Experience***	Time-saving Bias	
Reference Bias	Availability Bias	Parkinson's Law of Triviality	
Search Bias	Functional Fixedness	Well-travelled Road Effect	
Information Presentation Biases	Google Effect	***Biases related to Trends***	
Framing Bias	Law of the Instrument	Bandwagon Effect	
Similarity Bias	Mere Exposure Effect	IKEA Effect	
	Bias related to Trends	***Biases related to Previous Knowledge / Experience***	
	Bandwagon Effect	Habit	
		Law of the Instrument	
		Mere Exposure Effect	
		Negativity Bias	
		Biases related to strategy-making	
		Test Bias	
		Hyperbolic Discounting	
		Inconsistency	

Cognitive Biases Catalogue
An Example : Planning Fallacy

Planning Fallacy	
Definitions Block	
Definition 1: The tendency to underestimate task-completion times. Definition 2: The planning fallacy is a phenomenon in which predictions about how much time will be needed to complete a future task display an optimism bias and underestimate the time needed.	
OODA Class: Decide Phase	OODA Subclass: Complexity
Reasoning for classification: Time is a crucial factor in software projects. Often, the implementation times fall short of the initial estimates. The reason being underestimation of task-completion times due to lack of understanding of the complexities involved.	
Examples and impact on architecture design decisions	
Example: Choosing spring-security as the security framework: Spring is one of the most popular choice for developing Java-based enterprise applications. To meet the security requirements, spring-security would be an automatic choice as it is part of the framework itself. It is easy to assume that configuring the application security would be as easy as developing an application in spring. However, it is not an easy solution to implement without a proper understanding. If the decision-makers assumes that the security aspect is as easy as feature development, then it leads to an optimism bias resulting in time estimate errors. *Impact:* A common result is missing delivery deadlines. The added pressure resulting from the missed deadlines leads to implementation of sub-par solutions.	
Debiasing techniques	
The decision-maker must understand how to estimate time. There are many workflows for time estimation which can be used. One simple way is to add a buffer time to the initial time estimate in order to complete tasks. It is common to set the buffer time to 10% of the total estimate.	
Related biases	
Complexity bias, Parkinson's Law of triviality, Time-saving bias.	

Expert Reviews

- Feedback and reviews were gathered by presenting results online
  - Target group – software architects, lead developers and product owners

> “The biases and classification feel genuine. The question next is how to rectify them.“
> - Vice President and Software Architect a Morgan Stanley

> “We are biased because we take a look at what worked in the past and trust our future decisions based on it”
> - CTO at Schoen Digital Labs

> “Lots of biases and too much information. Reading all of it was intensive“
> - Software Architect at Siemens

> “It would be helpful if I could somehow get a notification as to which stage of decision-making I am in along with the biases I should be aware of“
> - Software Architect at Siemens

> “Content is good and worth reading“
> - Lead developer at J.P Morgan

Research Implications

- Enforcing a structured way of decision-making from software architecture perspective to make less biased decisions
- Avoid *observe* and *orient* paralysis
- Basis for developing cognitive bias recognition engine for decision support systems
- Trainings – companies such as Siemens and IBM provide basic trainings on cognitive theories

Thank you