DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master Thesis in Informatics

Decision-making and Cognitive Biases in Designing Software Architectures

Entscheidungsfindung und Kognitive Bias beim Entwurf von Softwarearchitekturen

Author: Akash Manjunath

Supervisor: Prof. Dr. Florian Matthes

Advisor: Manoj Mahabaleshwar M.Sc.

Date: February 21, 2018

I assure the single-handed composition of this master’s thesis only supported by declared resources.

Garching, 21.02.2018

(Akash Manjunath)

Acknowledgement

First and foremost, I want to thank my parents, Veena and Manjunath, for always being understanding and patient. My girlfriend, Sharada, for constantly supporting me at every step and ensuring that I maintained a high level of motivation throughout the thesis.

I want to thank Prof. Dr. Florian Matthes for providing me with the opportunity to pursue my Master Thesis at the SEBIS chair. My supervisor Manoj, who was highly supportive of me during the time of the thesis and provided a great environment to work in. The weekly discussions helped me challenge myself to deliver a quality thesis.

Further, I want to thank my friends Kavyaa, Prateek, Krishna, Akshay and Amith for their timely insights and support. Pushkar, especially, for helping me immensely during my days here in Germany.

Lastly, I am grateful to all the people who provided their valuable feedback during the evaluation phase as it played a crucial part in upgrading the quality of the thesis content.

Abstract

The architecture of any software can be thought of as a blueprint about its structure. This blueprint is the end artifact generated based on a series of decisions taken by software architects and decides the overall quality of the resultant software. Software architects, being human, are invariably subject to the influence of cognitive biases during their decision-making due to cognitive limitations of the human mind. This results in a systematic deviation from the ideal decision-making process leading to sub-par solutions because of missing rationality behind the decisions.

This thesis focusses on identifying and formalizing decision-making models(DMM) in the context of making software architecture design decisions(SADD). There are two approaches towards decision-making, namely *normative* and *behavioral*. Previous studies present various DMMs under both approaches. Within the scope of this thesis, three models are investigated in detail within this thesis: *the rational economic model*, *the bounded rational model* and *the recognition-primed decision(RPD) model (naturalistic decision-making framework)*. Each step of the DMMs is mapped to the *OODA Loop* (Observe, Orient, Decide and Act) decision cycle. Additionally, different types of cognitive biases relevant to making SADD are identified and classified under one or more phases of the OODA Loop. Detailed information about each bias is documented and presented as part of a cognitive bias catalogue and a brief explanation on how to use it is presented.

The main target group of this thesis is software architects responsible for making SADD. Software architects possess contrasting experiences and the differences in their experiences is a decisive factor in decision-making. For example, experienced architects could use the RPD model during their decision-making using past experiences as a basis for the decision-making. Junior architects, on the other hand, could use RPD model due to their enthusiasm for innovation and follow latest trends. The primary goal is to make decision-makers understand how the different DMMs, the OODA Loop, and the various cognitive biases are related to each other and about their potential impact during decision-making. The increased awareness equips the decision-makers with ample information to make rational decisions with less bias.

Contents

List of Figures

List of Tables

1 Introduction

1 Introduction

Software development life cycle defines a sequence of steps to elicit requirements, design, implement, test, deploy and maintain software systems. The requirements elicitation phase produces requirement documents. The follow up step is the design phase which forms the foundation on which the implementation, testing and maintenance of the software takes place. The end artifact of the design phase is the software architecture of the system which encapsulates the set of decisions made by software architects.

1.1 Problem statement

Designing software systems involves continuous decision-making and is an iterative process. Different decision-making models (DMMs) can concretely represent this process. Every DMM is comprised of a series of steps specifying the course of actions taken to reach a decision. Furthermore, each DMM has its own set of advantages and limitations. The limitations of the DMMs adversely affect quality of the resultant software.

This thesis focuses on one specific limitation when making software architecture design decisions(SADD) – the cognitive limitation of software architects leading to the designing of sub-par software architectures.

The cognitive limitation is due to the limited capacity of the human brain in dealing with complexity and manifests itself in the form of cognitive biases. By definition, a cognitive bias is a systematic pattern of deviation from rationality in judgement [17]. Cognitive biases limit objective reasoning resulting in biased decision-making. Thus, there is a need to avoid cognitive biases or at least reduce their impact to make quality SADD.

1.2 Core Concepts

To understand the different cognitive biases influencing the decision-making, three main concepts are investigated during the course of the thesis. First is the concept of the OODA (Observe, Orient, Decide and Act) Loop decision cycle. This is explained in detail in the first section of Chapter 2. It is a popular decision-making tool originated from a military background and presently used by decision-makers in other fields as well. The four phases of the OODA Loop are represented as four verticals in the skeleton matrix in Figure 1.

The second area of focus is the decision-making process itself and its formal representation through DMMs. The different DMMs considered during the research is presented in section 2 of Chapter 2 in the decision-making section. Three relevant models from the context of SADD are investigated in detail in sections 3, 4 and 5 of Chapter 3. Each of the three DMMs is presented as a sequence of steps through separate process models and are the three horizontals in the skeleton matrix respectively as shown in Figure 1.

The three horizontals and the four verticals are intersected to establish a relationship between the two concepts as shown in the following figure:

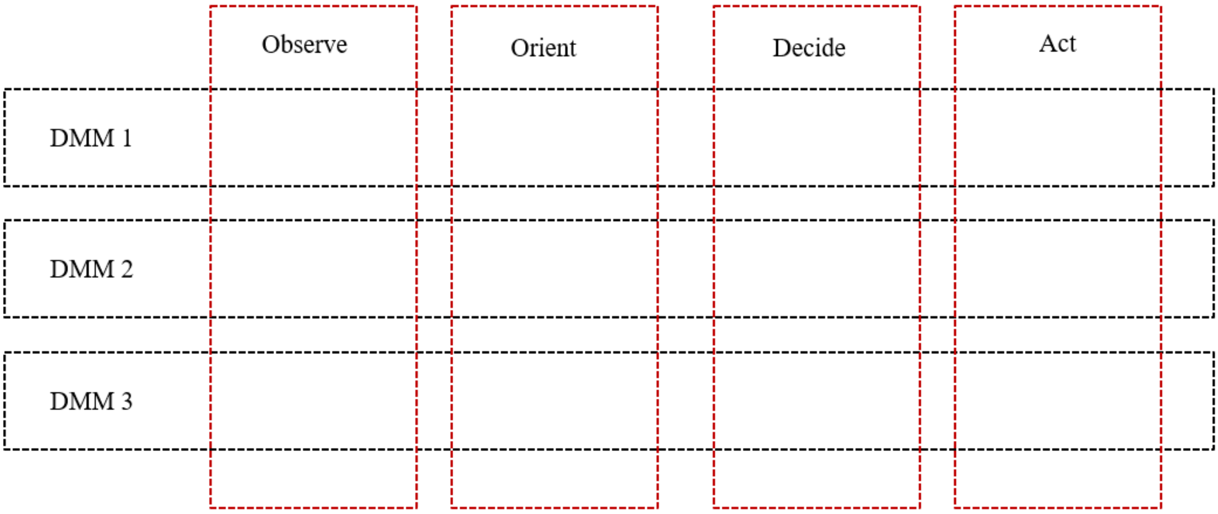


Figure 1: Skeleton matrix of DMMs and the OODA Loop phases

The third and final area of research is on cognitive biases and its different variations. A list of cognitive biases is created by consolidating information from previous research and is part of Chapter 3, section 6. Appendix 1 contains the final list of over two hundred types of biases and their definitions. Out of the two hundred odd types, thirty-three were deemed as relevant for SADD and examined further. The biases are then classified in two levels. The first level classifies the cognitive biases under one or more of the observe, orient, decide or act phases. In the second level, a custom classification under each phase of the OODA Loop is made to further enhance understandability through increased modularity of information. The results are presented in section 8 of Chapter 3 in the form of a cognitive bias catalogue.

1.3 Research Goals

The goal of the thesis is twofold. The primary goal is to establish a relationship between the DMMs and the OODA Loop through the matrix representation and map the different cognitive biases to the different phases of the OODA Loop. This is to present the combination of the three concepts in a manner that can be understood by decision-makers. The information triggers the availability heuristic by making the information available thereby aiding in debiasing of decision-making to ensure sound rationality behind decisions. This leads to the enhanced quality of decisions resulting in better SADD.

The secondary goal is to help decision-makers to avoid the observe-orient paralysis. This is a common phenomenon observed during decision-making process wherein decision-makers are stuck in an observe-orient loop and are unable to proceed to the decide and act phases.

1.4 Research Questions

The following research questions are addressed in this thesis:

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?

1.5 Target Group

The research conducted in this thesis is focused towards the community of software architects. However, decision-making is a continuous process and often other actors such as software developers, testers, product owners and others are involved in making decisions which impact the quality of a software. Thus, the end results of the thesis could potentially be beneficial to all those actors involved in the SADD making process.

1.6 Using the Thesis Artifacts

A detailed description on how to use the artifacts of the thesis is presented in Chapter 5. The methodology suggested is one of many ways in which the artifacts can be adapted for use in a real-world scenario. However, the readers are free to use the artifacts as deemed fit according to given scenarios.

2 Related Work

2 Related Work

2.1 OODA Loop

The concept of OODA loop was first put to paper by military colonel John Boyd as a decision-making process for military purposes. OODA represents decision-making in four phases - Observe, Orient Decide and Act as show in Figure 2.1. It is a strategic tool in which an actor makes observations about the surrounding environment, orients his thinking process by perceiving the important information based on the context, decides on a course of action and finally implements the course of action. While it was first introduced in the military field, the concept has been successfully adapted in business and other fields.

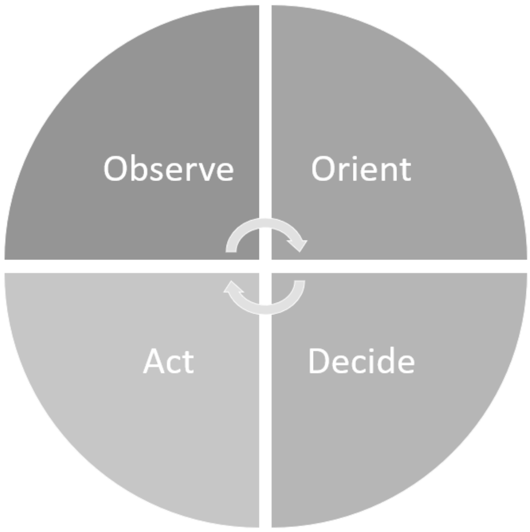


Figure 2.1: High Level Overview of the four phases of the OODA Loop Decision Cycle

2.1.1 The need for understanding the OODA Loop

Decision-making is generally an implicit process, in the sense that the decision-makers are not always aware of the exact steps of decision-making. The aim behind gaining a thorough understanding of the OODA Loop is to make explicit the implicit aspects of decision-making. Understanding it helps in dealing with ambiguity during decision-making. “Ambiguity is central to Boyd’s vision… We never have complete and perfect information…” [14]. In order to deal with ambiguity, decision-makers must shift perspective and update their “mental models”. The inability to do so results in continued work against outdated “mental models” and in the face of change, leads to failure.

Additionally, decision-makers experience the observe-orient paralysis when faced with ambiguity. The reason behind this can be understood by exploring the detailed view of the

OODA Loop as shown in Figure 2.2. The sense-making occurs in the orient phase, and in case the decision-makers fail to make sense of the information they tend to loop back to the observe phase to gather more information resulting in them being stuck.

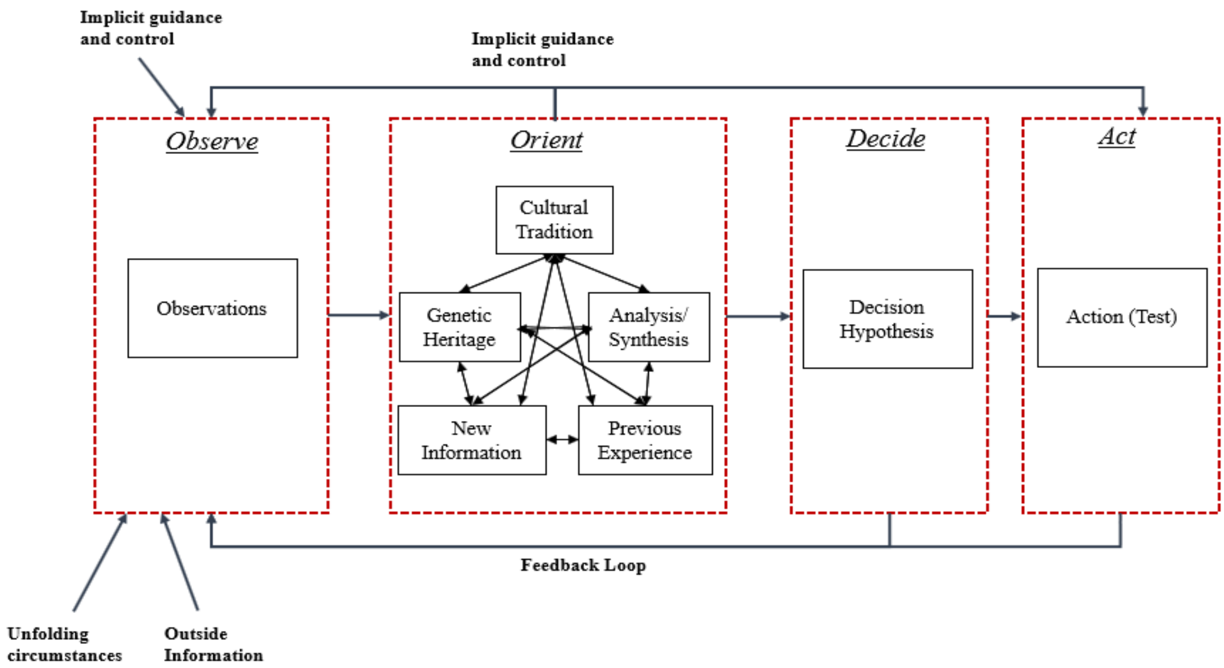


Figure 2.2: Detailed View of the OODA Loop (Adapted from the Journal of Defense Software Engineering)

To overcome being stuck in the observe-orient paralysis, the key is to be mentally agile. The response to being stuck is to conduct more observations. Due to the loss of time, the final decisions are then forced resulting in a lack of sound rationality behind them. Decisions made under such circumstances suffer from the "golden hammer" syndrome as the decision-makers are likely to make the same decisions based on previous experience to overcome every problem. This results in more resources consumed in an attempt to execute a failing strategy.

By understanding and being aware of the principles behind the OODA loop, one can deal with ambiguity by rapidly updating their mental models and avoid the observe-orient paralysis. From a software point of view, it prevents the accumulation of technical debt.

The following sections provide a detailed view about the four phases of the OODA Loop.

2.1.2 Observe Phase

The first phase of the OODA Loop is the observe phase. In this phase, the actor takes into account new information pertaining to the changes in the environment. This information is crucial in forming new mental models and to deal with confusion generated from ambiguity and uncertainty.

Boyd mentions two common problems encountered in the observe phase. The first problem is in dealing with incomplete information gathered from imperfect observations. The second one being the overabundance of information leading to difficulty in separating the actual information from the unwanted information.

The above problems or pitfalls can be tackled by developing a sound judgement, which is the main objective of the subsequent orient phase.

2.1.3 Orient Phase

The orient phase involves making sense of the information gathered in the observe phase.

According to Boyd, this is the most important step and he termed it as the “schwerpunkt”, a German word that translates roughly to “main point of emphasis” [18]. The reasoning behind this is the fact that the orient phase is responsible for shaping our mental models which then decides the course of actions.

The challenge of the orient phase is to continuously update the mental models in an environment that is rapidly changing. It involves breaking down of the older models through a process called as “destructive deduction” and constructing new ones through the process of “creative induction”. This is a continuous process of changing the mental model as soon as there is a change in the environment. The aim is to have a mental model which represents the current reality at all times. Boyd also mentions that having multiple mental models allows for better orientation resulting in better decision-making.

2.1.4 Decide Phase

According to Boyd, the decide phase is when the actor decides from among a set of alternatives generated from the orient phase. The selection of a perfectly matching mental model is near impossible due to imperfectness of information pertaining to the environment.

The “hypothesis” mentioned Figure 2.2 under the decide phase is because decision-makers hypothesize that a course of action can aid in addressing a concern and then and then create a mental model based on that hypothesis. The validation of the hypothesis is done in the final step, the act phase.

2.1.5 Act Phase

The last phase of the OODA Loop is the Act phase. Upon deciding on a mental model, the decision-maker must act on it. The act phase consists of a testing step to validate the hypothesis from the decide phase. "Feedback into the systems act as validity checks on the correctness and adequacy of the existing orientation patterns. [20]" This indicates that the OODA Loop is not only a decision process, but a learning system [14]. Through action, validation of the mental models takes place. By performing multiple actions and validating based on the feedback, the actor can decide on the best mental model for a given scenario.

2.1.6 Setting the tempo

"We got to get an image or picture in our head, which we call orientation. Then we have to make a decision as to what we're going to do, and then implement the decision…Then we look at the action, plus our observation, and we drag in new data, new orientation, new decision, new action, ad infinitum…" [14]. The effectiveness of decision-making lies in how quickly decision-makers can iterate over the OODA Loop and rapidly establish mental models.

Tempo is a crucial underlying element in implementing the principles of the OODA Loop. Therefore, understanding the importance of setting the tempo is critical. Not finding the right tempo results in "resetting" of the OODA Loop throwing teams into chaos and confusion. However, this does not imply that maintaining a rapid tempo is the way to using the OODA Loop effectively. Boyd specifies that making rapid changes in tempo by being fast or slow according to situations is the way to utilize the OODA Loop effectively.

2.1.7 Conclusion

To conclude, the OODA Loop makes explicit the implicit knowledge of decision-making process [18]. It also allows the manipulation and control of the decision-making process. The OODA Loop is a learning engine that allows an individual or organization to thrive in a changing environment. Decision-makers must constantly stay in touch with current industry trends and then update their mental models to exercise effective orientation.

2.2 Decision-making

In simple terms, a decision can be termed as a choice or a conclusion drawn from a piece of information. From a psychological perspective, decision-making is a cognitive process taking place within an actor's mind leading to the selection of a course of actions from among a set of alternative courses of actions.

2.2.1 Modeling the Decision-making Process

As stated in the previous section, decision-making is an implicit process. To make this implicit process explicit, the process of decision-making is represented through various process models. A decision-making process model is simply a sequence of steps representing the actions taken by an decision-maker which results in a decision being made. Researchers have presented several models over the years. Within the scope of this thesis, two approaches to decision-making are examined in detail in the forthcoming sections: the normative approach and the behavioral approach.

2.2.2 Normative Approach

The normative approach, also called as the rational approach, consists of a set of decision-making models which are based on sounds logical reasoning and statistics. It can be argued is mostly applicable in an ideal world scenario wherein a perfect requirements specification with no future changes is available, alongside other factors such as ideal time, budget, team and other factors in a software development project. There are three models representing the normative approach [8], namely the *Rational Economic model(REM)*, *Brunswick's Lens model* and the *Cynefin framework*.

2.2.2.1 Rational Economic Model

As the name suggests, this DMM comprises of a series of steps representing a rational process. The model is represented in Figure 2.3 and it is evident that the flow of this model is straightforward and logical. The first step in the model is for the actor to define the concern or the problem statement. Additional information is gathered about the concern to ensure that the definition is complete. Once the concern is defined, the actor proceeds to create a list of all existing alternatives that can potentially help in addressing the concern. In the next step, an *unbiased* ranking algorithm is applied to the alternative list to aid the selection process. On ranking the alternatives, an "optimal" alternative is chosen. The optimal alternative is then implemented and tested to verify its effectiveness.

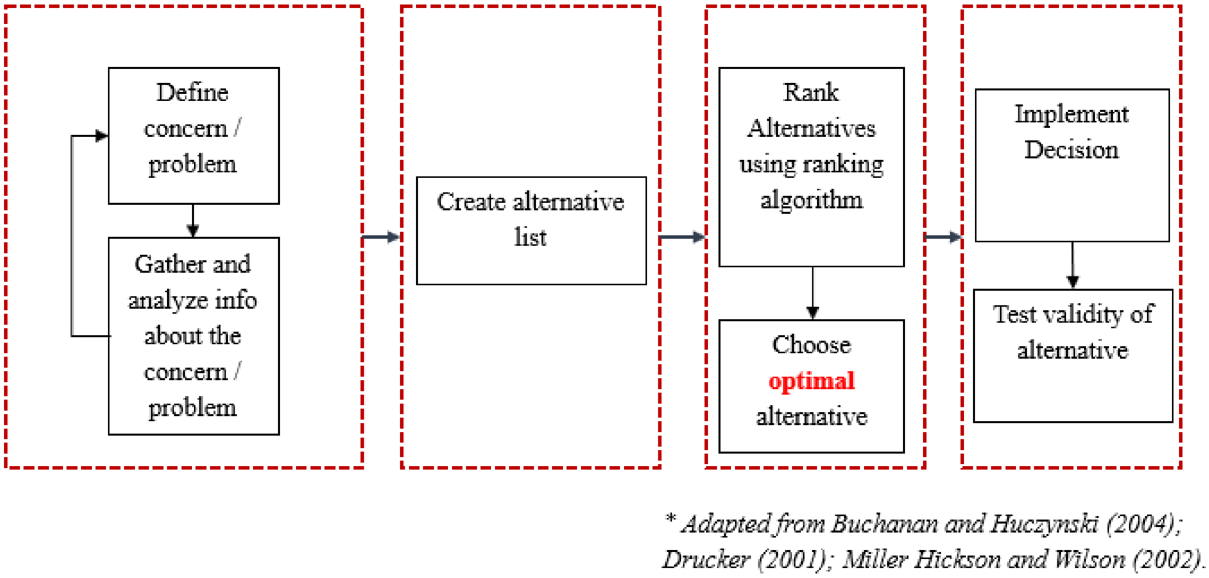


Figure 2.3: Rational Economic Model

2.2.2.2 Brunswick's Lens Model

The Lens model is based on a statistical point of view. It comprises of three elements: the basic information available in the situation where a decision must be made, the actual decision made by the decision-maker and the optimal decision which should have been made.

The first element which is the basic information sets the decision variables and forms the basis for decision-making.

The second element of the model is the actual decision made by the actor. Egon Brunswik (1952; 1956) stated that actors take two aspects into consideration during decision-making: the environment and the object of the decision. These two aspects are examined through different factors or cues which are context dependent. These cues decide the statistical weight to be assigned to each possible decision based on which the final decision is chosen.

The third and the final element of the model is the optimal decision. This represents the best possible course of action in a particular scenario. This decision exists in theory and is only possible in an ideal world scenario where all cues are "ultimate" in nature.

2.2.2.3 The Cyenfin Framework

The Cynefin Framework is designed for leaders and policy-makers to aid in their decision-making. "The framework links learning and knowledge"[1]. The model is designed around five core concepts or parameters – simple, complicated, complex, chaotic and disorder. It reflects the perceptions of people and how they make sense of what they perceive. By using the Cynefin Framework, decision-makers can view the environment from new perspectives allowing them to take different approaches towards problem solving.

2.2.3 Behavioral Approach

"Behavioral models refer to the way in which decisions are actually made"[2]. The behavioral approach assumes that there never exists an ideal world scenario when making decisions. The models falling under this are generally the approaches that decision-makers experience in real world scenarios. The behavioral approach takes into consideration the fact that all decisions made by decision-makers occur under cognitive limitations. "It was found that people behave differently in 'real world' situations than they do in 'laboratory' conditions ."[3] They involve less logic and are less structured when compared to the normative or rational approach. There are three models falling into the behavioral approach [8] - *Incrementalism*, *Naturalistic Decision-Making* and *Bounded Rationality*. The models are examined in the subsequent sub-sections.

2.2.3.1 Incrementalism

In the Incrementalism model, the decision-maker does not make huge strides in attempting to solve a problem. In complex scenarios, a decision cannot be made in one go. "Instead, they(decisions) slowly evolve in a series of small incremental steps"[4]. The decision-maker uses experience and intuition to "muddle through" the steps and the outcomes of each step are carefully monitored. The advantage of using this model is that it ensures that decision-makers avoid serious mistakes when making decisions. This is especially useful for managers as small changes ensure evolution in the long run without compromising stability. An aspect of incrementalism is that defining concerns and generating the list of alternatives is seen as a

---

[1] Cronjé and Burger, 2006

[2] Hill, 1979b

[3] Duggan and Harris, 2001; Pruitt, Cannon-Bowers and Salas, 1997

[4] Tarter and Hoy, 1998

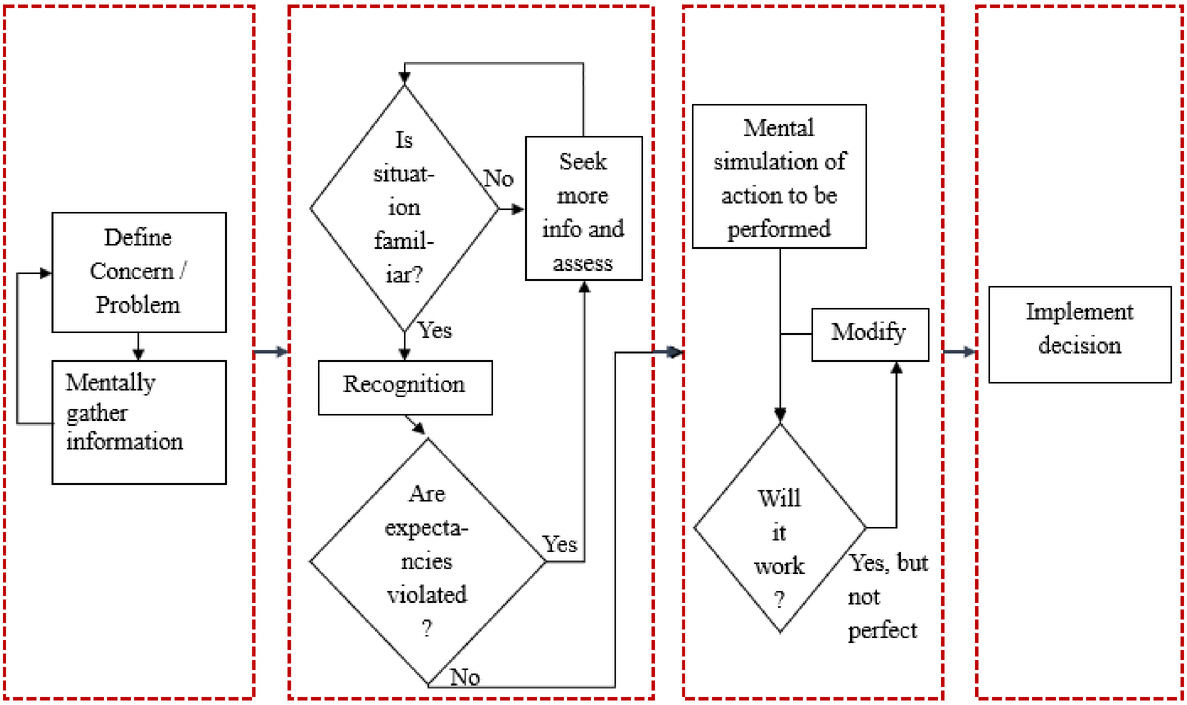
single step. Alternatives are not examined in a sequential manner to find an optimal one. In fact, the concerns change as the decisions evolve. The down side of using this mode is that it is less agile and cannot cope with sudden and rapid changes.

2.2.3.2 Naturalistic Decision-making

The Naturalistic Decision-making(NDM) is one of the most common framework of decision-making. This framework is largely used in scenarios where decision-makers are to make decisions under stressful conditions such as extreme time pressure and high stakes. This is also true when the decision-maker is relatively inexperienced.

Decision-makers do not generate alternative list and evaluate them under a standard evaluation criterion. They rely on mind maps to orient themselves with the scenario at hand to make decisions. The effectiveness of the decision is entirely dependent on the abilities of the decision-maker as it relies on mentally recognizing patterns from past experiences or otherwise and dealing with them.

The Recognition-Primed Decision (RPD) model is derived from the NDM framework its steps are presented in Figure 2.4. Initially, the decision-maker starts by defining the concern or the problem. Information is gathered mentally drawn from past experiences and the decision-maker defines the concern until it is adequate. Once the concern is adequately defined, the decision-maker makes a mental assessment as to whether the situation is familiar on the basis of past experiences. If the situation is familiar, the decision-maker proceeds to verifying if any expectancies are violated. In the face of any abnormalities or unfamiliar situations, additional information is gathered. Once the situation is familiar, then a mental simulation of the action is made and checked to see if it works. If the situation is not familiar, then the action is modified to check if it works when evaluated mentally. The final step is to implement the decision.



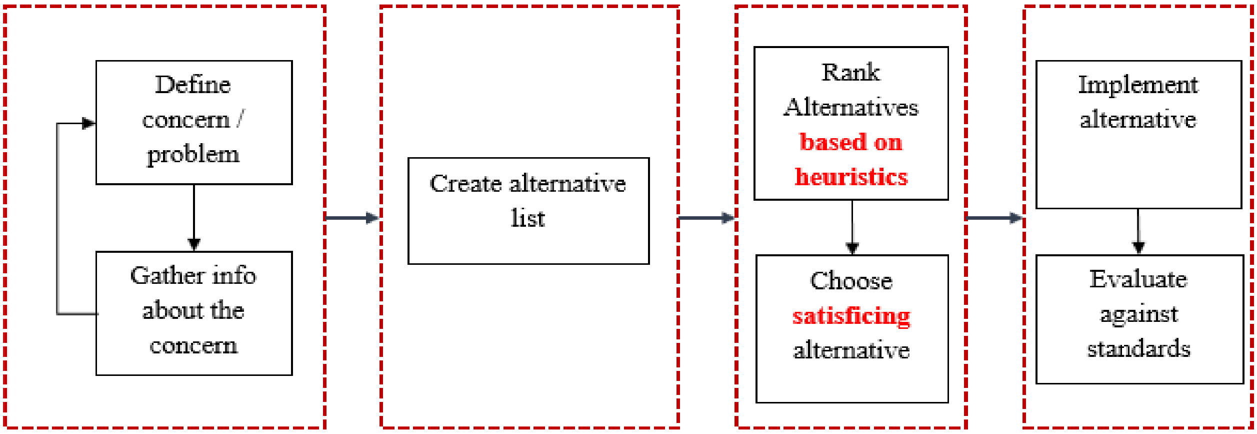
* *Adapted from : A Recognition Primed Decision (RPD) Model of Rapid Decision Making by Gary Klein (1993)*

Figure 2.4: Recognition Primed Decision Model

2.2.3.3 Bounded Rationality

As opposed to the normative approach, in a real-world scenario, decision-makers often list down alternatives and examine them in a sequential manner. When dealing with a complex scenario or in a situation where using a normative approach is not feasible due to time constraints and other pressures, "decision-makers look for the first workable alternative. [8]" The selected alternative may not the optimal one and does not necessarily satisfy all the conditions required to address the concern. Such an alternative is termed as the "satisficing alternative" as depicted in figure 2.5. Once a satisficing alternative is found, the decision-makers do not expend additional effort in examining other alternatives. This is the Bounded Rational model of decision-making [5]. This model is one of the most influential decision-making models as well.

Bounded Rationality is centered around the decision-maker constructing simplified mental models to deal with a given concern. The first stage is similar to the RPD model wherein the decision-maker defines the concern by gathering information. The only difference in this case is that the information is not only gathered from mentally from past experiences, but also from additional sources like other team members, books and other sources depending on the time constraints. Once the concern has been defined, an alternative list is created in the next stage. The selection process stage is made up of two steps consisting of ranking the alternatives and choosing an alternative to act upon. In this model, the ranking is done based on heuristics such as past experiences, tendencies of the decision-maker such as having a bias for innovation and so on. Once the ranking is complete, the first workable alternative or the satisficing alternative is selected. The final stage involves implementing the alternative and evaluating against standards to verify is the chosen alternative was a viable one.



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

Figure 2.5: Bounded Rational Model

2.2.4 Cognitive Biases

By definition, a cognitive bias is a systematic pattern of deviation from rationality in judgement [17]. It results in decision-makers suffering from a loss in judgement leading to prejudiced decision-making. Such decisions lack in quality, especially in cases where the decision-maker is unaware of the influence of cognitive biases.

The reasoning behind the influence of cognitive biases can be attributed to the cognitive limitations of the human mind. Real world scenarios are time boxed and decision-makers use heuristics or mental shortcuts while making decisions.

Heuristics are based on intuition and is a simplifying strategy aiding decision-makers in dealing with complex scenarios. However, they can also result in a cognitive bias leading to incorrect assessments and a mismatch between judgement and reality. For example, anchoring and adjustment is one of the most common biases which often makes decision-makers rely more than necessarily on the first information that is found. Subsequent information found may not be taken into consideration seriously and might be eliminated without reasoning.

Thus, cognitive biases can be thought of as a logical fallacy derived from heuristics often leading to mental automation of decisions without sound rationality behind them.

2.2.4.1 Types of Cognitive Biases

Cognitive biases manifest themselves in different types in different people based on circumstance. Researchers have identified many types of biases over the years. Within the scope of this thesis, the list of cognitive biases has been aggregated from two main sources. The first one is from the Wikipedia page on cognitive biases which has an aggregated list of one hundred and eighty-six cognitive biases from researchers over the years. The second major source is from the paper "Cognitive biases and decision support systems development: a decision science approach" by David Arnott which lists thirty-seven types of cognitive biases. The entire list of two hundred and twenty-two biases along with their corresponding definitions is presented in Appendix 1.

3 Thesis Contribution

3 Thesis Contribution

3.1 OODA Loop and DMMs

In the previous section, the OODA Loop decision cycle was described as a popular tool referenced by decision-makers during the decision-making process. Additionally, two approaches of decision making - normative and behavioral approaches were presented. Under the normative approach, three models were described in detail: Rational Economic model (REM), Brunswick's Lens model and the Cynefin framework. Additionally, three models were described in under the behavioral approach as well: Incrementalism, Recognition-Primed Decision(RPD) model (Naturalistic Decision-making Framework) and Bounded Rationality. However, the possibility of linking the concepts of OODA loop and DMMs is relatively unexplored, especially from the context of making SADD. The following sections explain how the two concepts are combined to produce DMMs incorporating the concept of OODA Loop.

3.2 Selection criteria for DMMs

The first criteria for selection or elimination of DMMs is whether the DMM is viable in the context of making SADD. The reason being that software architects are the main target group within the scope of the thesis. The second criteria is the ease with which a DMM can be represented as a sequence of steps because the focus is on qualitative analysis and not quantitative analysis.

Under the normative stream, REM is selected. The Brunswick's Lens model is not considered as it is a model with a statistical background and is hard to represent as a sequence of steps. Moreover, the model requires the "ideal" decision against which the actual decision can be compared with. This involved additional complexities and is not within the scope of the thesis. The Cynefin framework is also not considered as it is designed for leaders and policy-makers to aid in their decision-making. Such decision-making is relevant from an organization perspective but does not influence SADD.

Two models under the behavioral stream - Recognition-Primed Decision(RPD) model (Naturalistic Decision-making Framework) and Bounded Rationality are considered for further analysis. The Incrementalism model is primarily designed for mitigating risks and is hard to depict as a sequential model and hence not chosen.

The next sections present the chosen DMMs and how they are related to the OODA Loop decision cycle.

3.3 Rational Economic Model

The first model investigated is the RE model. It was described previously as the DMM which is suitable in an *ideal-world* scenario. The aim in is to arrive at an *optimal* decision.

The model adaptation is made keeping in mind that software architects are the decision-makers. It starts with the software architect and a set of documents containing the functional and non-functional requirements from the requirements engineering phase. The software architect uses the documents to design the architecture of the software through a series of SADD.

During the decision-making process, the OODA Loop decision cycle is applicable starting with the observe phase. As the model deals with an ideal-world scenario, the end goal of the observe phase is to completely define the concerns or the problems. This is dependent on the completeness of the requirement documents and additional information about requirements is gathered in case of incompleteness. Once the concerns are defined, they are analyzed thoroughly and in the orient phase the software architect decides about the selection of rejection of potential alternative which can be used to address the concern. A list of alternatives is produced by the end of the orient phase. In the next phase, a decision must be made to choose an optimal alternative. For this an algorithm is used to rank the alternatives based on suitability. The assumption in the RE model is that there exists an ideal world algorithm capable of ranking the alternatives in an unbiased way. The act phase which is the final one involves implementation of the chosen optimal alternative and testing of its validity.

As decision-making is a continuous process, it implies that the OODA Loop is constantly applied by the software architect until the final software is delivered. The feedbacks loops allow for improved decision-making during the software development life cycle (SDLC).

The complete adaptation of the RE model in relation with the OODA Loop from the context of SADD is presented in Figure 3.1.

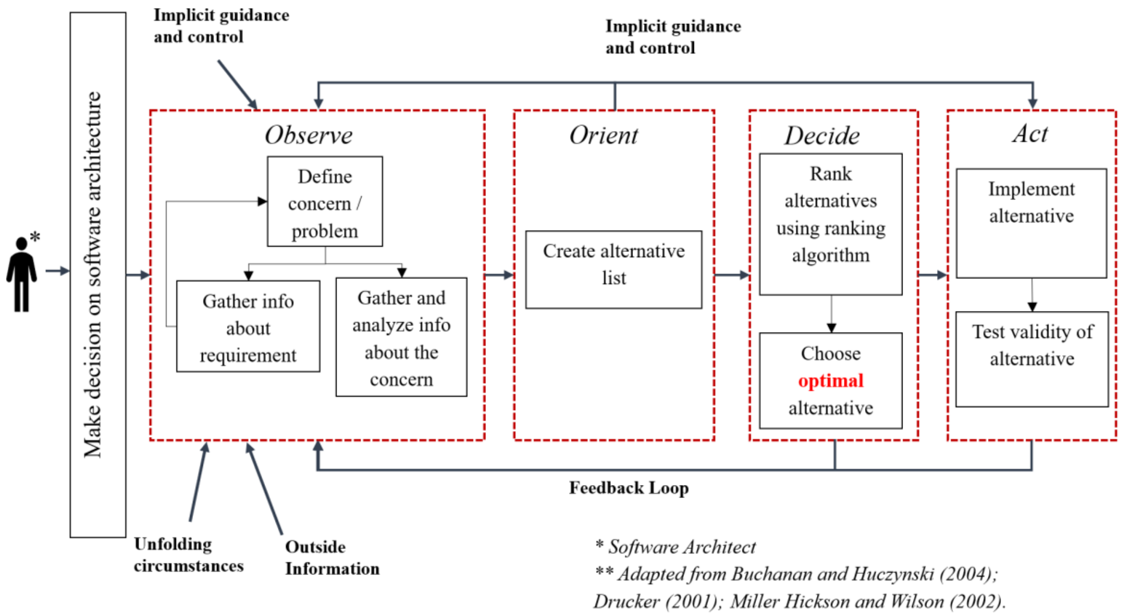


Figure 3.1 REM with the OODA Loop

3.4 Recognition-Primed Decision Model

The second model chosen is the RPD model. It falls under the framework of naturalistic decision-making and is heavily reliant on the cognitive abilities of the decision-maker. Since the RPD model falls is based on the normative approach, the model is reflective of a real-world scenario. This implies that software architects often deal with changing requirements and this needs constant changes in the design of the software architecture. Figure 3.2 presents the RPD model adapted to the OODA Loop.

As in the RE model, the RPD model begins with the conclusion of the requirements engineering phase. The software architect must use the requirement documents to make a set of SADD to design the software architecture.

The goal of the observe phase is to define the concerns or the problems as completely as possible. As the requirements may not be complete, the software architect mentally gathers information from past experiences and uses the knowledge make sense of the incomplete information.

In the orient phase, the situation is assessed to verify whether it is familiar or not. If the situation is familiar, then mental checks are performed to check if any expectancies are violated. In case of any unfamiliarity or any expectancies being violated, the software architect reorients by seeking more information. Once the situation is familiar, decisions are made on possible alternatives which can solve the problems and an alternative list is created.

Once the alternative list is generated, mental simulations of the alternatives are made to check for feasibility and an alternative is decided upon. The final decision results in the selection of either a *satisficing* alternative or a *sub-optimal* alternative depending on the cognitive capabilities of the decision-maker.

The chosen alternative is then implemented and evaluated against standards. The evaluation results are passed back as feedback to the observe phase for the next iteration of the OODA Loop.

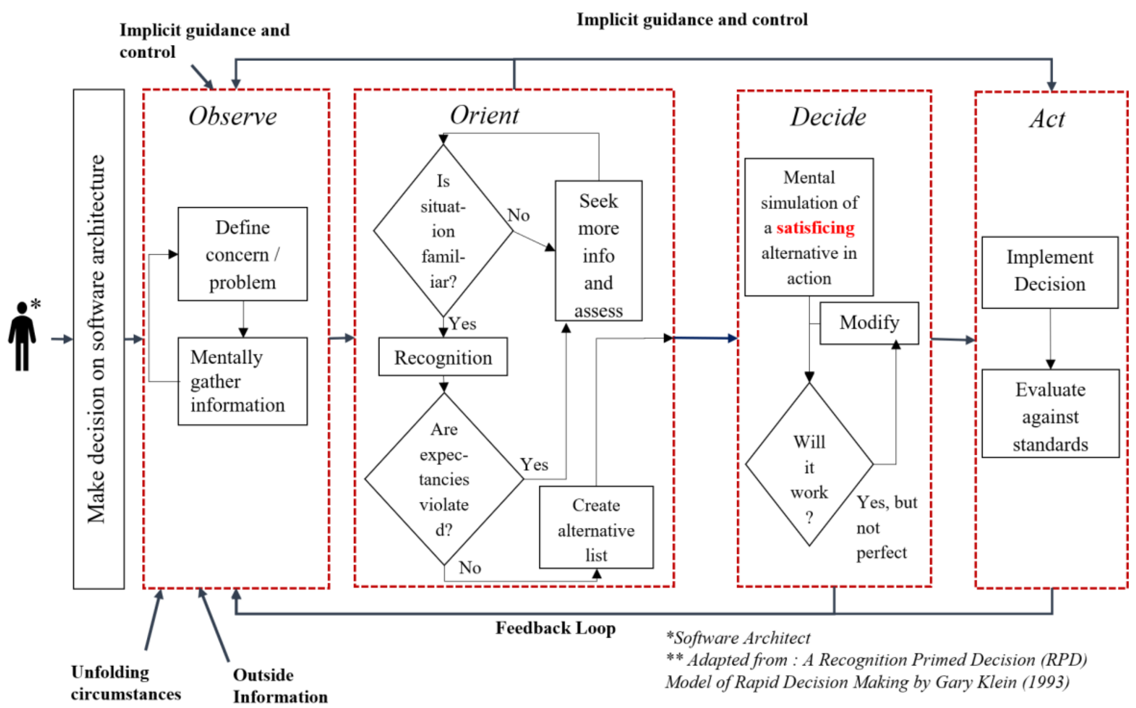


Figure 3.2 Recognition-Primed Decision Model with the OODA Loop

3.5 Bounded Rational Model

The final DMM analyzed is the BR model. It lies between the RE model and RPD model and can be applied in a real-world scenario as it is classified as a normative approach. Figure 3.3

shows the BR model adaptation with the OODA Loop. The BR model is quite similar to the RE model and can be thought of as the RE model being used under the limitations of real world scenarios.

Like all the other models, the BR model starts with the software architect having to design a software by making SADD based on the requirement documents from the requirements engineering phase.

The observe phase involves defining the concerns or the problems by making observations and gathering information. The only difference between the RE model and the BR model is that there is no feedback provided to refine incomplete requirements. Thus, the software architect defines the concerns using requirements which may be incomplete.

The orient phase is the same as with the RE model as the software architect makes sense of the concerns and decides on which alternatives could be used to address them.

The major difference is seen the decide phase because of the non-existence of an unbiased algorithm or system which can rank the alternatives. In the absence of such a system, the software architect ranks the alternatives based on cognitive heuristics such as past experiences, personal preferences, qualifications of the team and so on. The use of cognitive heuristics results in the choosing of a satisficing alternative which has the potential to be a good enough solution.

Finally, the chosen alternative is implemented and evaluated against standards. It is important to note that the feedback from the evaluation phase has an impact on which cognitive heuristics will be used by the software architect in the subsequent iterations of the OODA Loop.

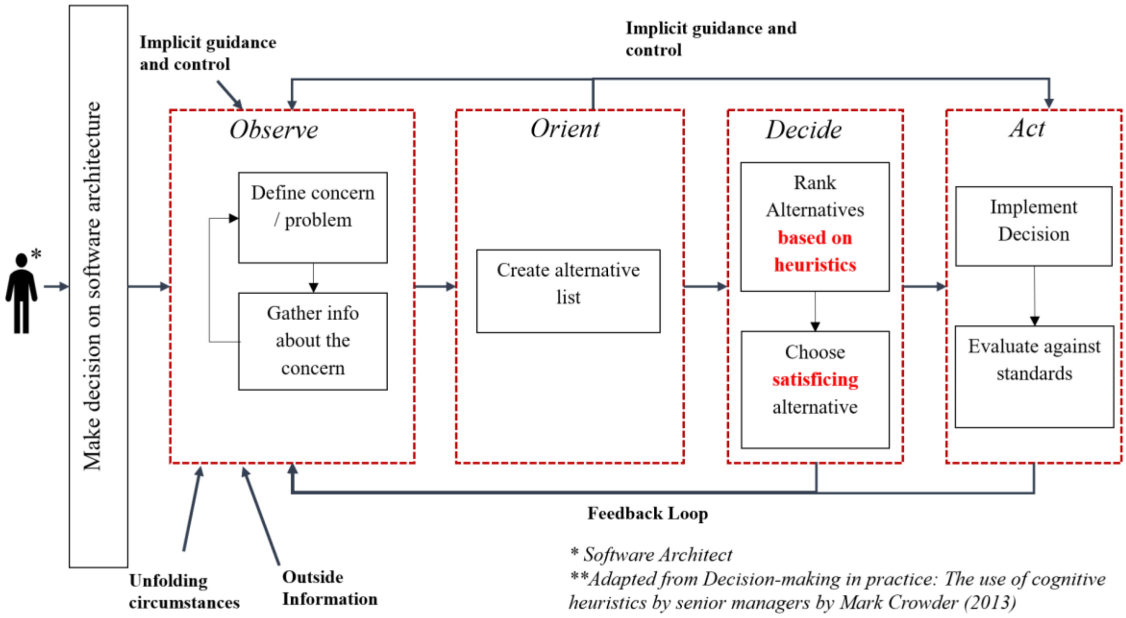


Figure 3.3 Bounded Rational Model with the OODA Loop

3.6 Cognitive Biases in the context of SADD

Software architects make SADD on a constant basis. As with every human decision-maker, software architects are subject to the influence of cognitive biases. Therefore, it is important for them to be aware of the different types of cognitive biases and the impact of each one on the decision-making process.

Initially, around two hundred and twenty cognitive biases were identified from different sources. However, not all of them are relevant for making SADD. Thirty-three of them were found to be relevant based after context, scope and time constraints during the thesis. These are presented in Table 3.1. The cognitive biases are then classified as described in the next sections.

Cognitive Biases in the context of SADD	
Attenuation	Law of the Instrument
Availability	Levels-of-processing Effect
Bandwagon Effect	Mere Exposure Effect
Base Rate Fallacy	Misinformation Effect
Completeness	Negativity
Confirmation	Parkinson's Law of Triviality
Framing	Planning Fallacy
Functional Fixedness	Post-purchase Rationalization

Google Effect	Reference
Habit	Search
Hard-easy Effect	Similarity
Hyperbolic Discounting	Test
IKEA Effect	Time-Saving Bias
Inconsistency	Well-travelled Road Effect
Information	

Table 3.1 Cognitive Biases from the Context of SADD

3.7 Classification of Cognitive Biases

A two-level classification of the selected cognitive biases is made. In the first level, every cognitive bias is assigned to one or more phases of the OODA Loop as in figure 3.4.

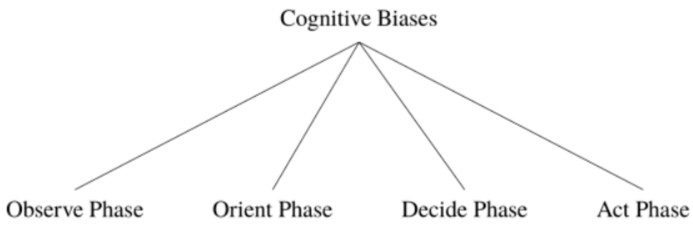


Figure 3.4 First level classification of cognitive biases using the phases of OODA Loop

The observe phase was further classified into two categories – information gathering biases and information presentation biases as depicted in Figure3.5. Information gathering biases are those biases which influence the decision-maker when gathering information. The information presentation biases are those which are related to how the information is presented to the decision-maker. For example, how the requirements are framed has an effect on how the decision-maker interprets the information. The cognitive biases related to information gathering are listed in Table 3.2 and comprise of those biases which influence the decision-maker's strategies while gathering information.

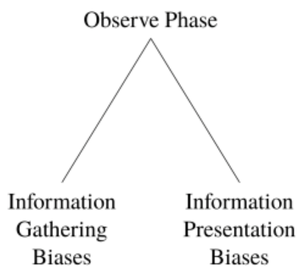


Figure 3.5 Sub-classification of Cognitive Biases in the Observe Phase

Cognitive Biases related to Information Gathering	
Completeness	Levels-of-processing Effect
Confirmation Bias	Reference Bias
Information Bias	Search Bias

Table 3.2 Cognitive Biases related to Information Gathering

Apart from gathering information, the observe phase also includes biases which are related to how information is presented. From the point of view of SADD, this relates to how information is presented in the requirement documents. The phrasing of sentences influences the interpretation of information in the orient phase. Table 3.3 shows the list of cognitive biases related to information presentation.

Cognitive Biases related to Information Presentation	
Framing Bias	Similarity Bias

Table 3.3 Cognitive Biases related to Information Presentation

The Orient Phase has four sub-classifications as shown in figure 3.6. The sub-classification is based on how information from the observe phase is interpreted by the decision-maker.

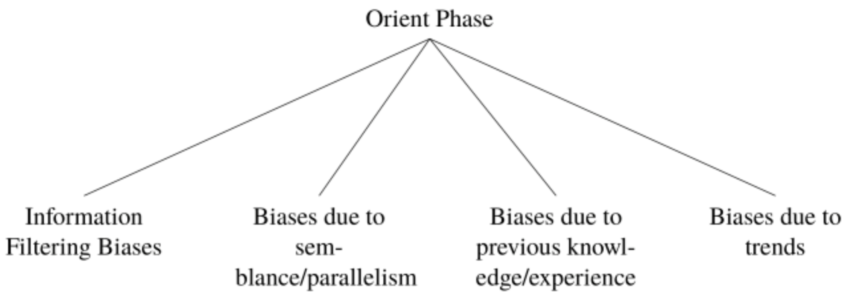


Figure 3.6 Sub-classification of Cognitive Biases in the Orient Phase

When the decision-maker interprets the information in the orient phase, it is common to filter information when the information is too much. The cognitive bias influencing this filtering process is presented in Table 3.4

Cognitive Biases related to Information Filtering
Base-rate Fallacy

Table 3.4 Cognitive Bias related to Information Filtering

The second sub-classification is related to similarity or dissimilarity of information. The cognitive bias related to semblance is shown in Table 3.5.

Cognitive Biases related to Semblance
Similarity Bias

Table 3.5 Cognitive Bias related to Semblance

The third sub-classification is related to the decision-maker's previous knowledge or experience. Decisions made in the past affect decision which will be made in the future. The cognitive biases in table 3.6 are related to the past decisions.

Cognitive Biases related to Previous Knowledge / Experience	
Availability	Law of the Instrument
Functional Fixedness	Mere Exposure Effect
Google Effect	

Table 3.6 Cognitive Biases related to Previous Knowledge / Experience

The final sub-classification is related to current trends and the cognitive bias related to this is shown in table 3.7.

Cognitive Biases related to Trends
Bandwagon Effect

Table 3.7 Cognitive Biases related to Trends

For the Decide Phase, there are four sub-classifications as well shown in figure 3.7. It has the most number of cognitive biases as the actual decision-making occurs in this phase. The decisions are made based on complexity of the problem, nature of how the solution will be invented, previous experience and strategy making.

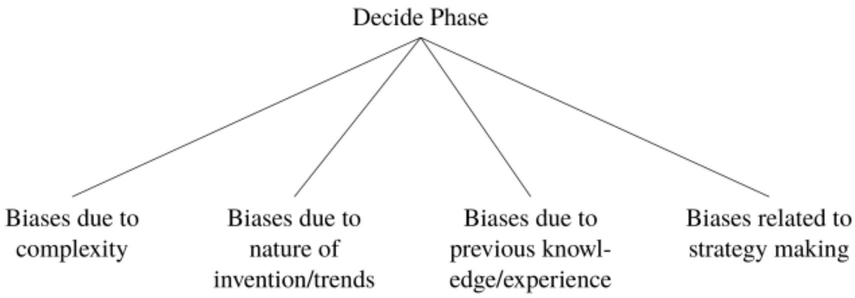


Figure 3.7 Sub-classification of Cognitive Biases in the Decide Phase

The first sub-classification in the decide phase is related to complexity. The cognitive biases which come into play based on the complexity of information available or the situation at hand is documented in table 3.8.

Cognitive Biases related to Complexity	
Attenuation	Time-saving Bias
Hard-easy Effect	Parkinson's Law of Triviality
Planning Fallacy	Well-travelled Road Effect

Table 3.8 Cognitive Biases related to Information Presentation

The second sub-classification is related to previous knowledge and experiences. This sub-classification is relevant in both the orient and the decide phase. The cognitive biases related to previous experience of the decision-maker and which influence the actual decision-making are presented in table 3.9.

Cognitive Biases related to Previous Knowledge / Experiences	
Habit	Mere Exposure Effect
Law of the Instrument	Negativity Bias

Table 3.9 Cognitive Biases related to Previous Knowledge / Experiences

The final sub-classification under the decide phase is related to the strategy making of the decision-maker. The strategies adopted depends on the nature of the decision-maker such as whether or not the decision-maker is risk averse, preferences for long-term or short-term planning and so on. The cognitive biases related to this are shown in table 3.10.

Cognitive Biases related to Strategy	
Hyperbolic Discounting	Test
Inconsistency	

Table 3.10 Cognitive Biases related to Strategy

There are no further classifications for the act phase since there are only two biases under it.

The next section if the cognitive bias catalogue which contains detailed description of each of the cognitive bias mentioned in the above tables.

3.8 Cognitive Bias Catalogue Template

The two-level classification of the cognitive biases relevant when making SADD break down the long list of cognitive biases into smaller modules. This way, the decision-maker needs to be aware of only a subset of the cognitive biases depending on which stage of decision-making is applicable. For example, if the decision-maker is in the observe phase, the being aware of just those cognitive biases in the observe phase is enough to reduce their impact.

Each cognitive bias in the previous sections is described in detail in the next section as part of the cognitive bias catalogue. The template of the catalogue is presented in table 4.18

<Bias Name>	
Definitions Block	
Definition:	
OODA Class:	Subclass:
Reasoning for classification:	
<reasoning>	
Examples and impact on architecture design decisions	
<examples>	
Debiasing techniques	
<techniques>	
Related biases	
<bias1, bias2…>	

Table 3.11 Cognitive Bias Catalogue Template

Every cognitive bias has a table consisting of six sections. The first section is the definition(s) block. As it states, it contains the definition of the cognitive bias from one or more sources. The second section mentions the classification and the sub-classification of the cognitive bias as described in the previous sections. In the third section, the reasons behind the classification of the cognitive bias is described. The fourth section presents one or more examples from real world scenarios are presented and potential impact on the software architecture is explained. The fifth mentions some debiasing techniques which can potentially aid in avoiding or reducing the impact of the cognitive bias. The examples are quite simple as it is intended for decision-makers with varying levels of experience. The final section mentions related cognitive biases.

3.9 Cognitive Bias Catalogue

This section consists of the cognitive bias catalogue. The catalogue is made up of four sections with each section consisting of those cognitive biases related to one of the four phases of the OODA Loop.

3.9.1 Cognitive Biases related to the Observe Phase

3.9.1.1 Completeness Bias

The completeness bias induces the feeling of completeness or incompleteness of information within the decision-maker. Gathering of further information depends on how many 'observations' have been made. The bias directly impacts the search for additional information and can create a false sense of the gathered information being complete enough to move to the orient phase.

<table>
<tr><td colspan="2">Completeness Bias</td></tr>
<tr><td colspan="2">Definitions Block</td></tr>
<tr><td colspan="2">The perception of an apparently complete or logical data presentation can stop the search for omissions.</td></tr>
<tr><td>OODA Class: Observe Phase</td><td>OODA Subclass: Information Gathering</td></tr>
<tr><td colspan="2">Reasoning for classification: Reasoning for class and subclass assignment: Completeness is related to how the information is 'observed' and whether further information will be gathered or not depending on the observation. The bias directly impacts the search for additional information especially in cases where there is a false sense of the gathered information being complete enough to move to the orient phase.</td></tr>
<tr><td colspan="2">Examples and impact on architecture design decisions</td></tr>
<tr><td colspan="2">Example: NoSQL database selection: Mongo DB is one of the most popular NoSql databases. Its popularity brings a tendency for software architects to decide on using it without gathering complete information about alternatives such as Couchbase, ArangoDB etc. This makes the selection pool for architecture decision incomplete.

Impact: Leads to illogical elimination of potential alternatives. It can also induce a feeling of incompleteness leading to the software architect spending more time on gathering more information without proceeding to the orient phase.</td></tr>
</table>

Debiasing techniques
In the above example, the choice of MongoDB was made in a ‘naturalistic’ way. To ensure a rational selection process, creating a selection pool of candidate technologies and decide on a set of parameters which can help in comparing their pros and cons for better decision-making.

Table 3.12 Completeness Bias

3.9.1.2 Confirmation Bias

The confirmation bias makes the decision-maker gather additional information to confirm what is already known. This adds no further value and does not provide the decision-maker with facts to contradict the already known information and provide an alternative point of view. Minimal effort is put into seeking information outside cognitive boundaries.

Confirmation Bias	
Definitions Block	
Often decision-makers seek confirmatory evidence and do not search for disconfirming information.	
OODA Class: Observe Phase	OODA Subclass: Information Gathering
Reasoning for classification: Confirmation bias leads decision-makers to observe information to confirm the information which they already possess. Minimal effort put into seeking information outside cognitive boundaries. Bias results in gathering of additional information to confirm what is already known.	
Examples and impact on architecture design decisions	
Example: NoSQL database selection: A software architect with prior knowledge about MongoDB would gather information to solidify claims for usage. This comes at the expense of gathering information about alternatives which could yield some disconfirming evidence on using MongoDB. *Impact:* Alternatives are not explored to the full extent to support a more rational decision-making. Additional time is spent on gathering evidence to support a decision which is already made.	

Debiasing techniques
While it is natural to seek additional information about known things, better decisions can be made by gathering equal data about unknown things as well. It is imperative that the decision-maker steps out of the comfort zone of known information.

Table 3.13 Confirmation Bias

3.9.1.3 Information Bias

Information Bias leads to unnecessary additional 'observations'. Software architects believe that gathering additional information can lead to better decision-making in spite of all the information being already gathered.

Information Bias	
Definitions Block	
Belief that furtherly acquired information generates additional relevant data, even when it evidently does not.	
OODA Class: Observe Phase	OODA Subclass: Information Gathering
Reasoning for Classification: Information Bias leads to additional unnecessary 'observations'. Architects believe that gathering additional information can lead to better decision-making despite all the information being already available.	
Examples and impact on architecture design decisions	
Example: NoSQL database selection: In the situation where enough information is available about NoSQL alternatives like Aerospike, Apache Ignite, MongoDB, Couchbase, FoundationDB, Oracle NoSQL DB, Redis etc., software architects sometimes still tend to look for more information. *Impact:* The additional information gathered adds no value and ends up wasting precious resources.	
Debiasing techniques	

When enough information is available about alternatives, move to orient and decide phases instead of being paralyzed in observe phase. Loopback to observe phase and make iterations in case there is a need for extra information.

Table 3.14 Information Bias

3.9.1.4 Levels-of-processing Effect

The levels-of-processing effect presents itself when software architects mentally gather information from past experiences when making future decisions. The quality of the mentally gathered information depends on how it has been 'encoded' into memory based previously. The quality of 'encoded' information depends on factors such as the sources from where it was gathered, whether the decision-maker has hands-on experience or only theoretical knowledge and so on.

Levels-of-processing Effect	
Definitions Block	
That different methods of encoding information into memory have different levels of effectiveness.	
OODA Class: Observe Phase	OODA Subclass: Information Gathering
Reasoning for Classification: Framework selection to develop a frontend application: In the current technology scenario where there are so many web frameworks such as Angular, React, EmberJS, VueJS among others. The choice of framework selection for a use case is made based on the extent to which the architect was able to process the effectiveness of the framework when being used.	
Examples and impact on architecture design decisions	
Example: Selection of a framework to develop a frontend application: Based on current trends, there are many frontend frameworks such as Angular, React, EmberJS, VueJS and so on. If a software architect has previously worked with Angular, then it would heavily influence the future decision. Whether or not the Angular is chosen again depends on whether or not the architect was able to effectively process all the necessary information about the framework into memory.	

Impact: The cognitive limitations of the architect plays a huge role in this case. If the information encode into memory was positive, then it is likely that Angular will be the choice in the future irrespective of whether other frameworks have more potential or not.
Debiasing techniques
Evaluate the technologies which come to mind outside the boundaries of previous experience. In case the previous experience was positive, then try to find cons in using the framework. This gives added perspective during decision-making. Additionally, broaden the alternative list to include those technologies which don't come to mind as well.

Table 3.15 Levels-of-processing Effect

3.9.1.5 Reference Bias

Software architects obtain information from multiple sources. Based on the quality and relevance of the source, a reference point is established thereby setting the tone and influencing further steps of decision-making.

Reference Bias	
Definitions Block	
The establishment of a reference point or anchor can be a random or distorted act.	
OODA Class: Observe Phase	OODA Subclass: Information Gathering
Reasoning for Classification: In the observe phase, information is obtained from different sources. There are multiple sources from which information can be obtained from. Based on the quality and relevance of the source, a reference point is established which sets the tone for the further steps of decision-making. A start point is set from where to gather information thereby influencing the decision-making process chain.	
Examples and impact on architecture design decisions	
Example: Influence of external information sources: Generally, when designing the architecture for a new software, architects conduct some research and gather as much information as possible. Every architect has specific sources from which they gather information such as Google, Stack Overflow, colleagues, communities for specific	

technologies and so on. The information obtained establishes a reference point for the next steps of decision-making. *Impact:* Quality of the final decision depends on how well a reference point was established initially. It becomes hard to erase the initial reference and evaluate future information in an unbiased manner.
Debiasing techniques
Establish formal parameters when gathering information from multiple sources based on a given context. Compare the information obtained using the parameters and justify the selection of a source as a valid reference point. This should be documented as decision logs for future references.
Related Bias
Anchoring and Adjustment

Table 3.16 Reference Bias

3.9.1.6 Search Bias

The goal of the observe phase is to create a knowledge base for decision-making. Software architects employ different search techniques for this purpose. One such search technique is to use information which contains a high frequency of relevant buzz words. This creates an illusion that the information found is highly relevant for the given scenario.

Search Bias	
Definitions Block	
An event may seem more frequent because of the effectiveness of the search strategy.	
OODA Class: Observe Phase	OODA Subclass: Information Gathering
Reasoning for Classification: The goal of the observe phase is to create a knowledge base for decision-making. Architects employ different search techniques for this purpose. Tendency is to gather information from sources which have higher frequency of certain search keywords which creates the illusion that the information being highly relevant in the context, even if it is not.	

Examples and impact on architecture design decisions
Example: Gathering information to select a technology for backend: Arguably, Java and NodeJS are the most popular backend technologies. However, there are other technologies such as Go, Scala, Django, Ruby on Rails, Symfony and others which could be viable alternatives. When using search engines like Google, the information obtained depends on the usage of the search key terms. For example, searching for 'high performance web application framework' results in a direct link to Symfony being listed in the first page of Google results. While other links mention competing technologies, direct links to their home pages are noticeably missing from the first page. *Impact:* Software architects with limited experience rely heavily on the information that can be found readily on the internet. Thus, the architecture decisions would then be driven by search results without having a rational basis according to a given scenario.
Debiasing techniques
Use different sources such as books, forums and other avenues to search for information. Document the search terms used by different members to analyze the search strategies used as part of decision logs.

Table 3.17 Search Bias

3.9.1.7 Framing Bias

Software architects avail information from various sources. Some common ones include requirement documents, technology documents presented by sales personnel of the companies, short descriptions found via Google search and so on. Much effort goes into the framing of such information and the manner of presentation has an impact on the decision-makers.

Framing Bias	
Definitions Block	
Definition 1: The framing effect is an example of cognitive bias, in which people react to a choice in different ways depending on how it is presented. Definition 2: Events framed as either losses or gains may be evaluated differently.	
OODA Class: Observe Phase	OODA Subclass: Information Presentation

Reasoning for Classification: Information is available to architects from various sources Requirements document from the requirements engineering phase, technology documents presented by sales personnel of the companies are some examples. All such information with different levels of framing bias is gathered during the observe phase. The framing of requirements influences presentation of information.
Examples and impact on architecture design decisions
Example1: Framing of requirements document: The starting point for architects is generally the requirement documents and they gather information based on the documents. The manner in which the requirements are framed influences the understanding of what needs to be done. *Example2: Advertisements about technical solutions:* Advertisements from companies are framed to influence the end user into using their products. 'Loss Framing', 'Gain Framing', and 'Statistical Framing' are generally associated with advertisements. They are designed to create a positive feeling about a technology and to create a negative feeling about their competitors. *Impact:* In the first situation, it is important for a software architect to understand the requirements correctly to design the right solution. Otherwise, the resulting solution does not match the requirements. In the second situation, it is often the case that companies are 'duped' into buying solutions thinking that it meets their requirements.
Debiasing techniques
In the first case, the person framing the information documents must keep the readers in mind and in what ways the information can be interpreted. Sufficient communications must be held with the architects and the people responsible for framing the requirements to make the clarify the them. In the second case, when buying external solutions or investing in third party technologies, one must try and create a set of parameters against which different solutions can be compared with. Getting reviews, trial periods and conducting rigorous proof-of-concepts are some ways to avoid getting into the framing bias trap.
Related Bias

Mode (The mode and mixture of presentation can influence the perceived value of data), Similarity Bias

Table 3.18 Framing Bias

3.9.1.8 Similarity Bias

Past decisions always influence future ones. If a software architect feels that the current scenario is similar to a previous one, then the outcome of the previous decision impacts the current decision. If the previous outcome was positive, then the decision-maker will tend to make the same decision and vice-versa.

Similarity Bias	
Definitions Block	
The likelihood of an event occurring may be judged by the degree of similarity with the class it is perceived to belong to.	
OODA Class: Observe Phase	OODA Subclass: Information Presentation
Reasoning for Classification: Gathering of information is usually based on how much a person. Information at hand can seem similar to previous information due to the way in which it is presented.	
Examples and impact on architecture design decisions	
Example: Setup a static website: Setting up a static website may not require a database connection. However, in some cases, a database connection might be setup simply because the requirement sounded similar to previous one which required a working database. *Impact:* Due to similarity of information, architects tends to take the same decisions as they did in the past when presented with similar information. This may or may not be the right solution as there is less likelihood of two scenarios being the same.	
Debiasing techniques	
Treating every scenario as a different use case will result in exploration of different technologies. Use the new information to compare with knowledge gained from previous experiences to make more informed decisions in the future.	

Related Bias
Framing

Table 3.19 Similarity Bias

3.9.2 Cognitive Biases related to the Orient Phase

3.9.2.1 Base Rate Fallacy

The information gathered from the observe phase is subject to processing and interpretation during the orient phase. It is common to filter out and focus on specific information during the orient phase due to reasons such as too much information, time pressure, cognitive limitations and other reasons. Base rate fallacy makes the decision-maker focus on information which feels specific to the scenario and ignore generic information under the assumption that they are unimportant.

Base Rate Fallacy	
Definitions Block	
The tendency to ignore base rate information (generic, general information) and focus on specific information (information only pertaining to a certain case).	
OODA Class: Orient Phase	OODA Subclass: Information Filtering
Reasoning for Classification: The information gathered from the observe phase is subject to processing and interpretation during the orient phase. It is common to filter out and focus on specific information during the orient phase due to various reasons such as too much information, time pressure, cognitive limitations and other reasons.	
Examples and impact on architecture design decisions	
Example: Microservice vs Monolith architecture: When deciding between a microservice and a monolith architecture, the tradeoffs must be carefully considered. For example, from the view point of reliability, microservices have an advantage over monolith since the failure of one microservice will not bring down the entire setup. *Impact:* In the above example, if the decision is to be taken purely based on the reliability factor alone, then microservices is the way forward. However, when setting up a software architecture, there are multiple factors aside from just reliability such as availability,	

complexity, management, deployment and others. Microservice architectures are more complex to setup than monolith architectures. Decisions taken by purely focusing on just a few aspects might have an adverse effect in the future.
Debiasing techniques
Ensure that all the parameters to make a rational comparison are listed down initially. Rank the parameters based on the requirements which then sets up a sounds rationality for the decide phase.
Related biases
Anchoring and Adjustment, Focusing effect

Table 3.20 Base Rate Fallacy

3.9.2.2 Similarity Bias

Similarity bias is relevant in the context of orient phase when two similar alternatives are at hand. The superficial information about the alternatives are similar making it difficult for a software architect to make a rational decision on which one to choose.

Similarity Bias	
Definitions Block	
The likelihood of an event occurring may be judged by the degree of similarity with the class it is perceived to belong to.	
OODA Class: Orient Phase	OODA Subclass: Semblance / Parallelism
Reasoning for Classification: Similarity bias is relevant in the context of orient phase when information at hand feels similar to information obtained from a previous use case. As an architect, the tendency is to automatically interpret the information in the same manner due to the similarity to make design decisions.	
Examples and impact on architecture design decisions	
Example: Version control systems: There are plenty of version control systems such as Github, GitLab, BitBucket etc. Each option varies slightly in comparison with one another. Consider the case where a choice between GitLab and BitBucket is to be made. Until a few months ago, choosing Gitlab would have been an easy choice to make based on the parameter	

of continuous integration. Gitlab provides its own continuous integration. However, BitBucket added the pipelines features very recently as its own continuous integration feature. This makes Gitlab and BitBucket viable options and the decision to choose one is not so straightforward now since both options are quite similar. *Impact:* In the above example, take into consideration the use case of setting up continuous integration to run multiple independent steps in parallel. If the decision is to use BitBucket due to it being similar to GitLab, then this use case would be difficult to implement as the Pipelines feature of BitBucket does not support this feature yet whereas GitLab already has the feature implemented.
Debiasing techniques
Seek more information to distinguish the current requirements with similar ones to avoid applying the same solution simply due to similarity. When superficial information is not enough to clearly distinguish between alternatives, loop back to observe phase and gather additional information.
Related biases
Distinction (The tendency to view two options as more dissimilar when evaluating them simultaneously than when evaluating them separately)

Table 3.21 Similarity Bias

3.9.2.3 Availability Bias

In naturalistic decision making, architects rely on mental shortcuts which when evaluating the information gathered from the observe phase. The mental shortcuts result from their years of experience.

Availability Bias
Definitions Block
Definition 1: The tendency to overestimate the likelihood of events with greater "availability" in memory, which can be influenced by how recent the memories are or how unusual or emotionally charged they may be.

Definition 2: The availability heuristic is a mental shortcut that relies on immediate examples that come to a given person's mind when evaluating a specific topic, concept, method or decision.	
OODA Class: Orient Phase	OODA Subclass: Previous knowledge / Experience
Reasoning for Classification: In naturalistic decision making, architects rely on mental shortcuts which when evaluating the information gathered from the observe phase. The mental shortcuts result from their years of experience.	
Examples and impact on architecture design decisions	
Example: NodeJS frameworks – Hapi vs Express: Hapi and Express are both Node.js web application frameworks providing a robust set of features for building applications and services. Both are quite similar with minute differences. Some sources claim that Hapi is more effective as compared to Express when dealing with large teams to enforce conventions for code maintainability. Consider the case wherein the decision-maker is in charge of a large team but has recently worked with Express. *Impact:* In the above example, the decision-maker tends to choose Express due to greater availability of information in memory about it as compared to Hapi. This results in selection of Express over Hapi despite it being better suited for the given scenario.	
Debiasing techniques	
Counting on previous experiences and mental shortcuts to design solutions is part of naturalistic decision making. As with any bias related to experience, it is important to make sure that the interpretation from a previous scenario is applicable to the current scenario and to justify it so as to move to bounded rationality.	

Table 3.22 Availability Bias

3.9.2.4 Functional Fixedness

Often, architects prefer to rely on tried and tested methodologies to design solutions. This is especially true in the case of highly experience architects with less preference towards innovation to stick to well-known strategies based on their previous experiences.

Functional Fixedness	
Definitions Block	
Limits a person to using an object only in the way it is traditionally used.	
OODA Class: Orient Phase	OODA Subclass: Previous knowledge / Experience
Reasoning for Classification: Often, architects prefer to rely on tried and tested methodologies to design solutions. This is especially true in the case of highly experience architects with less preference towards innovation to stick to well-known strategies based on their previous experiences.	
Examples of occurrences	
Example: Developing a desktop application: There are many different technologies to develop desktop applications. Traditionally, languages such as C#, Objective C, Swift are used to develop native desktop applications for Windows and Mac OS as the languages exists for such a purpose. This means that development and maintenance of two code bases needs to take place. However, in cases where applications which do not require access to native drivers can be developed using other technologies. For developing JavaFX is a reasonable alternative for building desktop applications. Additionally, the emergence of Electron has made the development of container applications for desktop has become a reality. Applications can be developed using frameworks such as Spring, Django etc. and packaged with Electron as desktop applications. *Impact:* Technologies are constantly evolving due to ever changing requirements. This implies that using a technology need not be constrained to how it was meant to be used in a traditional sense. Sometimes simple solutions can be developed by using an existing technology innovatively instead of investing resources in developing more complex solutions through other technologies.	
Debiasing techniques	
Do not always associate certain use cases with specific technologies. Keep in mind the existing technology stack and the skill sets of the team and conduct some proof of concepts to verify if the existing technology can be leveraged to tailor new solutions.	

Related biases
Law of the instrument

Table 3.23 Functional Fixedness

3.9.2.5 Google Effect

Search engines are becoming increasingly powerful. Irrespective of the domain of work, people are reliant on search engines such as google for information. The ease of obtaining information has a downside – people tend to forget the easiest of things leading to what some people term as 'digital amnesia'. Based on the requirement documents, software architects use their previous experiences to come up with solutions. Due to cognitive limitations of the mind, it is not always possible to remember all the information and the tendency to forget is more profound due to the influence of search engines. Search engines cannot have all the answers and downside is that judgements are often anchored to the information presented by the search results.

Google Effect	
Definitions Block	
The tendency to forget information that can be found readily online by using Internet search engines.	
OODA Class: Orient Phase	OODA Subclass: Previous knowledge / Experience
Reasoning for Classification: Search engines are becoming increasingly powerful over time. Irrespective of the domain of work, people are reliant on search engines such as google for information. The ease of obtaining information has a downside – people tend to forget the easiest of things leading to what some people term as 'digital amnesia'. Based on the requirements document, architects use their previous experiences to come up with solutions. Due to cognitive limitations of the mind, it is not always possible to remember all the information and the tendency to forget is more profound due to the influence of search engines. Search engines cannot have all the answers and judgements are often anchored to the information presented by the search results.	
Examples and impact on architecture design decisions	

Example: Forgetting how one previously coded a solution: Due to the Google Effect, we tend not to commit data to memory simply because we know that it can be found online. This is quite true in cases when writing some complex code snippets. If the snippet can be found online, the easiest solution is to reuse the code and often developers do not bother to understand the logic behind it as long as there is a working solution. *Impact:* Based on the example, a dependency is developed on the source from where the solution was copied. When a similar situation presents itself in the future, the tendency would be to search online for the code snippet instead of being able to code it themselves. It also results in complacency in developers as they fail to double check if the code is applicable and passes all the border test conditions. It is true for software architects as well as they tend to forget the information about alternatives previously since it can be found again. However, there is always a risk of being unable to find that source again and information gained previously could be lost.
Debiasing techniques
Using more than one search engine would provide more perspective as each search engine has their own way of ranking search results. Some basic documentation consisting of decision logs, crucial information gathered when making a previous decision and some first level analysis will be helpful in reducing dependency on search engines.

Table 3.24 Google Effect

3.9.2.6 Law of the Instrument

Decision-maker often rely on familiar tools and do not explore new ones as long as the tried and tested tools work. This leads to a "golden hammer" approach of the decision-makers always using the same "instruments" for every use case even if it not the best tool available. With software architects, the tendency it to stick to tried and tested technologies as long as it works without effectively exploring new ones.

Law of the Instrument
Definitions Block
An over-reliance on a familiar tool or methods, ignoring or under-valuing alternative approaches.

OODA Class: Orient Phase	OODA Subclass: Previous knowledge / Experience
Reasoning for classification: As the definition suggests, this bias leads to an over-reliance on familiar tools and methodologies. The familiarity with tools is due to previous experience in using them which creates a comfort zone leading to a higher tendency to use them.	
Examples and impact on architecture design decisions	
Example: Selection of continuous integration tool: There are many tools for continuous integration available in the market. Jenkins being one of the most popular CI tools would be the preferred choice of many. Alternatives such as Travis, Bitbucket Pipelines, GitLab CI and others being are not considered in a fair way. *Impact:* Overly relying on tried and tested methodology leads to a one size fit all approach which does not work in every context. Moreover, it restricts innovation as new technologies with different approaches are not considered.	
Debiasing techniques	
As an initial step, conduct brainstorming sessions and create a list of alternatives based on inputs from all members of the team as well as some external sources. Rank the alternatives based on educated discussions. Based on the constraints such as time, pressure, team size and so on, conduct some proof of concepts to check feasibility of alternatives.	
Related biases	
Functional Fixedness, Mere exposure effect	

Table 3.25 Law of the Instrument

3.9.2.7 Mere Exposure Effect

This cognitive bias is similar to the law of the instrument bias. Software architects tend to favor approaches which they are more familiar with and alternatives are discarded without any rationality behind them.

Mere exposure effect
Definitions Block

The tendency to express undue liking for things merely because of familiarity with them.	
OODA Class: Orient Phase	OODA Subclass: Previous knowledge / Experience
Reasoning for classification: As the definition suggests, this bias leads to an over-reliance on familiar tools and methodologies. The familiarity with tools is due to previous experience in using them which creates a comfort zone leading to a higher tendency to use them.	
Examples and impact on architecture design decisions	
Example: Choosing a cloud computing service provider: AWS is arguably the biggest cloud computing service provider at the moment. When setting up a cloud based infrastructure for projects, AWS is the preferred choice for many simple because they are familiar with it. Alternatives such as Google Cloud, Microsoft Azure and others could be viable alternatives in many cases. One such case is using Kubernetes. Google has good support for it whereas AWS has recently rolled out a feature to support it. *Impact:* People prefer to try and use the same approach and technologies for all situations without rationally considering alternatives. In this case, a software architect with a preference for AWS will choose it again and discard Google. This would systematically eliminate the use of Kubernetes since the AWS support for it may not be as good. This, two good technologies are eliminated simply because the decision-maker was not familiar with it.	
Debiasing techniques	
As in the case of law of the instrument, the approach to debiasing would be to create an alternative list and rank them based on pros and cons. Depending on different constraints, quick proof of concepts can be done to aid in making the final decision.	
Related biases	
Functional Fixedness, Law of the instrument	

Table 3.26 Mere Exposure Effect

3.9.2.8 Bandwagon Effect

This is one of the most common cognitive biases. Software architects choose technologies because it is being used by most people. They may not deep dive into whether the technology will be a good fit for the scenario at hand because of blind belief in the technology arising from its large user base.

Bandwagon effect	
Definitions Block	
The tendency to do (or believe) things because many other people do (or believe) the same. Related to groupthink and herd behavior.	
OODA Class: Orient Phase	OODA Subclass: Trends
Reasoning for classification: External trends have an impact when a decision-maker is orienting based on the information at hand. The effect is more profound when there are a high number of alternatives or in cases where there is a tendency to use new technologies.	
Examples and impact on architecture design decisions	
Example: Javascript framework selection: This is one of the most common examples in the recent years. The emergence of numerous Javascript frameworks and libraries such as Angular, React, Vue, Ember and so on has led to a 'selection headache' for architects. *Impact:* While a large number of options to choose from is good to have, every new technology has a peak when the number of users spike up in a short time period. During the time frame, the general tendency would be to choose the options which is booming in the market and it makes it hard to make rational choices.	
Debiasing techniques	
Do not discard 'legacy' technologies just because of market trends. When it comes to new technologies, often the spikes in the number of users are short lived. Other factors such as future support, number of contributors, verifying if the framework can support all the requirements should be considered carefully before making decisions.	
Related biases	
Anchoring and Adjustment	

Table 3.27 Bandwagon Effect

3.9.3 Cognitive Biases related to the Decide Phase

3.9.3.1 Attenuation

When presented with too much information, it is normal for software architects to ignore some information to reduce the complexity and the amount of information to focus on. The decision-maker has to ensure that the information being ignored does not have any future value.

Attenuation	
Definitions Block	
A decision-making situation can be simplified by ignoring or significantly discounting the level of uncertainty.	
OODA Class: Decide Phase	OODA Subclass: Complexity
Reasoning for classification: From the definition, it can be inferred that discounting of information occurs when dealing with options which are commonly associated with high complexity. Less complex options have less uncertainty associated with them since all the details related to them is easily perceived. More complex options result in a lack of complete understanding leading to increased complexity and uncertainty and are often dealt with by making assumptions.	
Examples and impact on architecture design decisions	
Example: Capturing audio from external devices connected to desktops for live audio streaming: This is a common use case when it comes to audio streaming applications wherein sound has to be captured from external audio equipment connected to desktops. Some alternatives include Java Sound, Port Audio, MMDevice API and so on. Most of these options have the capabilities to capture sound from microphone and some basic audio equipment quite easily. However, when it comes to dealing with more complex devices with multiple channels, some of these options are not sufficient. *Impact:* The decision is quite tricky to make especially if the software architect lacks experience in the sound domain. The tendency would be to choose an option which may be able to handle most use cases. However, it may fail when it comes to handling more complex border cases.	
Debiasing techniques	

Ensure the chosen alternative is capable of handling all complex scenarios. The varying degree of uncertainty results in assumptions being made. It is imperative to make a list of such assumptions and perform some quick proof of concepts to validate the assumptions before making the decision to deep dive into the implementation.
Related Biases
Complexity

Table 3.28 Attenuation

3.9.3.2 Hard-easy Effect

Hard-easy Effect	
Definitions Block	
Definition 1: Based on a specific level of task difficulty, the confidence in judgments is too conservative and not extreme enough. Definition 2: The hard-easy effect is a cognitive bias that manifests itself as a tendency to overestimate the probability of one's success at a task perceived as hard, and to underestimate the likelihood of one's success at a task perceived as easy.	
OODA Class: Decide Phase	OODA Subclass: Complexity
Reasoning for classification: The hard-easy effect comes into play when options in the list of alternatives comprise of varying complexities. The general feeling is that by choosing a more complex alternative, it will yield in a higher success rate as compared to choosing an easier alternative.	
Examples and impact on architecture design decisions	
Example: Kubernetes for automated deployment and scaling of a website: With the rise of Docker around 2013, Kubernetes has gained traction in the recent years as the choice for automating deployments and scaling of container based applications. Kubernetes is a good choice when it comes to setting up complex websites requiring high availability and handling of large user bases. However, setting up the Kubernetes is not a simple task for someone with limited previous experience.	

Impact: The decision to go with a solution of high complexity need not always result in the best solutions. When dealing with more complex solutions, factors such as the skill set of the team, time constraints, feasibility of implementing the solution must be kept in mind. Sometimes, a simpler setup might result in the best possible solution. For example, when dealing with simple websites with a limited user base, a simple solution with Jenkins would probably suffice.
Debiasing techniques
The advantages and disadvantages of a complex versus simple solution must be clearly established. Long term visions, size and skillset of the team must be kept in mind to accommodate future requirements or varying complexities.
Related biases
Complexity bias

Table 3.29 Hard-easy Effect

3.9.3.3 Planning Fallacy

An important aspect of designing software architectures is the estimation of time required to implement it. Planning is of utmost importance as it decides whether or not the software can be delivered on time. Underestimation or overestimation of task-completion times is often the reason for failed software projects.

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.

Table 3.30 Planning Fallacy

3.9.3.4 Time-saving Bias

This is related to planning fallacy bias. The time-saving bias is common when decision-makers have to make choices under extreme time pressure or in cases where the resources at hand are extremely limited.

Time-saving bias	
Definitions Block	
The time-saving bias describes people's tendency to misestimate the time that could be saved (or lost) when increasing (or decreasing) speed.	
OODA Class: Decide Phase	OODA Subclass: Complexity

Reasoning for classification: Time misestimations often occurs due to lack of perception about the complexities involved. Decisions could be made on the presumption that increasing the factors such as the number of involved in the implementation of a project leads to saving time.
Examples and impact on architecture design decisions
Example: Assuming that adding more people to a late project results in faster implementation: In many projects, developers are promoted to a managerial position without proper training in project management. When dealing with sharp deadlines, such managers make the common mistake of adding extra workforce in a bid to meet the timelines. *Impact:* This is a classic case is based on Brooke's Law which states that "adding human resources to a late software project makes it later".
Debiasing techniques
There are exceptions to the above scenario. It is applicable only with respect to projects which are already late. Adding people to a project early on would be beneficial. Also, highly skilled contributors could be another exception as they would be able to contribute within a short time frame.
Related biases
Complexity bias, Planning Fallacy

Table 3.31 Time-saving Bias

3.9.3.5 Parkinson's Law of Triviality

Parkinson's law of triviality is when decision-makers assign importance to information which is not so important and vice-versa. In case of software architects, no so important parameters may be deemed as important and parameters which are actually important when designing the architecture could be ignored.

Parkinsons's Law of Triviality
Definitions Block
The tendency to give disproportionate weight to trivial issues. Also known as bikeshedding.

OODA Class: Decide Phase	OODA Subclass: Complexity
Reasoning for classification: Requirements are ranked based on their importance. The decisions which are made in relation to designing the architecture is directly dependent on which features are most important and the complexities involved in their implementation.	
Examples and impact on architecture design decisions	
Example: Deciding on feature implementation: During product development, sometimes an assumption is made that adding more features leads to product enhancement. The assumption for the need of every feature is not validated. This results in a software with a lot of features which may or may not be used by the users. *Impact:* Valuable time spent on developing features which are not used. The features have to be kept in mind while designing the architecture of the software thereby directly or indirectly affecting the design decisions.	
Debiasing techniques	
In the above case, validate the assumption that a particular feature is needed. Often, this occurs due to vagueness in the requirements document. In such a case, clarify the requirements before assigning weights to issues.	
Related biases	
Complexity, Planning Fallacy.	

Table 3.32 Parkinson's Law of Triviality

3.9.3.6 Well-travelled Road Effect

The well-travelled road effect is similar to planning fallacy. It creates a sense of familiarity leading to software architects underestimation the time required to deliver a software. In case of taking a less familiar route, the tendency would be to feel that additional time would be required for implementation irrespective of whether it is needed or not.

Well-travelled Road Effect
Definitions Block

Underestimation of the duration taken to traverse oft-traveled routes and overestimation of the duration taken to traverse less familiar routes.	
OODA Class: Decide Phase	OODA Subclass: Complexity
Reasoning for classification: After having ranked alternatives, the next step is to choose one for implementation. The tendency is to choose an option with which one is familiar with since the complexity involved is relatively lesser. An unfamiliar option is always associated with higher complexities leading to steep learning curve and exaggerated timelines.	
Examples and impact on architecture design decisions	
Example: Selection of a chatbot framework: 2017 has been the so-called year of chatbots. Natural language processing tools have made plenty of progress with options such as DialogFlow, IBM Watson, Lex and so on available in the market. All the above-mentioned technologies are quite similar in nature with subtle differences in terms of certain extra features or differences in under the hood implementation. Let us assume that the person in charge of making the selection for a new project is familiar with DialogFlow having worked with it previously. Due to familiarity with DialogFlow, the person would be inclined to choosing it again. *Impact:* This has an impact on the time estimates for implementation as it would be generally be lesser when using DialogFlow due to the feeling of it being a well-travelled road as opposed to when the decision is to go with an alternative like IBM Watson.	
Debiasing techniques	
In case of a familiar technology, do not underestimate the time to implement. Irrespective of familiarity, place a buffer of 10% if using a simple time planning strategy. In case of unfamiliar technologies, try to get an estimate from people well versed in working with those technologies and plan the time estimation strategies accordingly.	
Related biases	
Complexity bias, Time-saving Bias	

Table 3.33 Well-travelled Road Effect

3.9.3.7 Bandwagon Effect

In addition to being present in the orient phase, the bandwagon effect is present in the decide phase as well. As mentioned in the previous sections, software architects have a pool of potential alternatives in the decide phase from which they have to choose using some heuristics. One such heuristic is the tendency to be biases towards alternatives which are being used by most companies. Software architects end up choosing a technologies which is most commonly being used everywhere else, but it may not always be the right choice in a given context.

Bandwagon effect	
Definitions Block	
The tendency to do (or believe) things because many other people do (or believe) the same. Related to groupthink and herd behavior.	
OODA Class: Decide Phase	OODA Subclass: Nature of invention / Trends
Reasoning for classification: There is always a temptation to use trending technologies. When having to decide from a set of alternatives, the market trends often dictate the decision-making.	
Examples and impact on architecture design decisions	
Example: Choosing between Vanilla js and Typescript: There are plenty of programming languages based on javascript in the market. Typescript has been increasing in popularity and in the current market, the temptation would be to follow the trends. There is a tendency for people to migrate from vanilla js to typescript in case of existing projects or choose Typescript initially when implementing a new project. *Impact:* Using latest technology does not always ensure success. In case the technology is not supported well, or it has a strong competitor, then the technology may not survive. An example of javascript based language is CoffeeScript which has been facing a decline in spite of its popularity at one point.	
Debiasing techniques	
While it is important to keep up to date with the latest trends, it is important to remember that it does not always pay off. Keep track of competition and try to foresee which technology will have a longer lifetime based on factors such as the companies which are supporting the	

languages, the size of the contributing community and so on and make a rational decision instead of just following the trends.
Related biases
Pro-innovation bias (The tendency to have an excessive optimism towards an invention or innovation's usefulness throughout society, while often failing to identify its limitations and weaknesses.)

Table 3.34 Bandwagon Effect

3.9.3.8 IKEA Effect

The IKEA Effect is the tendency of software architects to be biased towards using technologies which have been developed within the organization. This develops a sense of protectionism towards using in-house solutions without fairly assessing external alternatives.

IKEA Effect	
Definitions Block	
The tendency for people to place a disproportionately high value on objects that they partially assembled themselves, such as furniture from IKEA, regardless of the quality of the end result.	
OODA Class: Decide Phase	OODA Subclass: Nature of invention / Trends
Reasoning for classification: A solution that is developed in-house will always be preferred within an organization over solutions developed by third party vendors. In many cases, the in- house solution will be preferred irrespective of whether or not it is feasible in the context due to the nature of its invention.	
Examples and impact on architecture design decisions	
Example: Choosing between Spring batch and an ETL product such as Informatica: One can argue that Java batch processing can be used to develop ETL solutions which are similar to ETL products such as Informatica. Both technologies have their own set of advantages and disadvantages depending on a specific context. *Impact:* In cases where companies have their own implementation using spring batch, the tendency would be to reuse the technology when a new requirement surfaces. The IKEA	

effect leads to a devotion to the software which has been developed in-house. Alternatives such as Informatica, Datastage etc. are not genuinely explored even in relevant use cases in the future because the decision-makers have a myopic view towards other alternatives.
Debiasing techniques
Do not decide on using a technology just because it is invented inside the company. The requirements must be kept in mind especially the ones which are meant to be implemented in the future. Additionally, the skill set of the team has to be kept in mind as to whether they are qualified to not only implement, but perform maintenance as well.
Related biases
Not invented here (Aversion to contact with or use of products, research, standards, or knowledge developed outside a group).

Table 3.35 IKEA Effect

3.9.3.9 Habit

Habit, in general, is a routine that is regularly repeated. When it comes to software architects, there is always a tendency to repeatedly make the same decisions. This is especially true if the decisions have yielded in positive results previously.

Habit	
Definitions Block	
An alternative may be chosen only because it was used before.	
OODA Class: Decide Phase	OODA Subclass: Previous knowledge / experience
Reasoning for classification: By definition, habit refers to an acquired behavior which is developed through knowledge gathered from different experiences.	
Examples and impact on architecture design decisions	
Example: Selection between AWS and Google Cloud for hosting a web application: AWS and Google Cloud are both good solutions for hosting a web application. In comparison to Google Cloud, AWS has been around for a longer period with more people are familiar with	

it. Given a scenario where a new web application is to be hosted, from habit a person with experience with AWS would prefer to use it again. *Impact:* While AWS has the advantage from having been around for a long time, Google Cloud is better suited in certain areas such as for big data applications. Making a decision simply because it was effective before does not result in the better choice.
Debiasing techniques
Making a list of advantages and disadvantages of alternatives is always helpful to compare. Inputs from people who have prior experience is always an advantage as they possess firsthand information from having used it.
Related biases
Law of instrument, Mere exposure effect

Table 3.36 Habit

3.9.3.10 Law of the Instrument

The law of the instrument is also referred to as the “golden hammer” concept. This is another cognitive bias applicable in the orient and decide phase. By law of the instrument, a software architect would decide on using the same technology stack for every use case irrespective of feasibility. It is similar to Habit.

Law of instrument	
Definitions Block	
An over-reliance on a familiar tool or methods, ignoring or under-valuing alternative approaches.	
OODA Class: Decide Phase	OODA Subclass: Previous knowledge / experience
Reasoning for classification: As the definition states, an exposure to a set of technologies for a prolonged period of time leads to an over-reliance on them. The excessive reliance is from experience gathered from using them over time.	
Examples and impact on architecture design decisions	

Example: Selection between AWS and Google Cloud for hosting a web application: AWS and Google Cloud are both good solutions for hosting a web application. In comparison to Google Cloud, AWS has been around for a longer period with more people are familiar with it. Given a scenario where a new web application is to be hosted, based on law of instrument a person with experience with AWS would prefer to use it again. *Impact:* While AWS has the advantage from having been around for a long time, Google Cloud is better suited in certain areas such as for big data applications. Making a decision simply because it was used before does not result in the better choice.
Debiasing techniques
Making a list of advantages and disadvantages of alternatives is always helpful to compare. Inputs from people who have prior experience is always an advantage as they possess firsthand information from having used it.
Related biases
Habit, Mere exposure effect

Table 3.37 Law of the Instrument

3.9.3.11 Mere Exposure Effect

The mere exposure effect is similar to habit and law of the instrument cognitive biases. Familiarity with a technology stack would result in a software architect deciding on the same options due to familiarity in using them.

Mere exposure effect	
Definitions Block	
The tendency to express undue liking for things merely because of familiarity with them.	
OODA Class: Decide Phase	OODA Subclass: Previous knowledge / experience
Reasoning for classification: Experience with technologies results in familiarity with them. The case of mere exposure effect occurs when a person has just heard or read about certain technologies being useful without having used it rigorously.	

Examples and impact on architecture design decisions
Example: Selection between AWS and Google Cloud for hosting a web application: AWS and Google Cloud are both good solutions for hosting a web application. In comparison to Google Cloud, AWS has been around for a longer period with more people are familiar with it. Given a scenario where a new web application is to be hosted. A person who has heard or read about Google Cloud would prefer to use it instead of AWS merely from exposure to such information. *Impact:* AWS and Google could have their own set of advantages and weaknesses. Making a decision without proper experience or feedback from experienced people and from mere exposure to information does not guarantee success.
Debiasing techniques
Making a list of advantages and disadvantages of alternatives is always helpful to compare. Inputs from people who have prior experience is always an advantage as they possess firsthand information from having used it.
Related biases
Habit, Law of instrument

Table 3.38 Mere Exposure Effect

3.9.3.12 Negativity Bias

The negativity bias is related to the previous experiences of software architects. If an architect had previously selected a technology and had an unpleasant experience in using it, then it creates a negative feeling towards using it again. The architect would discard such an option in the future even if there has been some good improvements later on.

Negativity bias
Definitions Block
Definition 1: Psychological phenomenon by which humans have a greater recall of unpleasant memories compared with positive memories. Definition 2: The negativity bias refers to the often-asymmetrical way we perceive the negative and the positive.

OODA Class: Decide Phase	OODA Subclass: Previous knowledge / experience
Reasoning for classification: Negativity bias arises knowledge gained from previous experience through the usage of technologies. In high-risk situations, it leads us to make intelligent decisions.	
Examples and impact on architecture design decisions	
Example: Adoption of Test Driven Development: There is no exact definition of how to use test driven development. It varies from team to team and from developer to developer. Due to ambiguity in the exact way of using it, it can prove to be quite tricky for developers to adopt this approach. If a software architect is trying to enforce a philosophy of high test coverage, then the developers could be forced into using the TDD approach. In case the developers do not come from a strong testing background, they may have to invest a lot of additional time to achieve such high test coverages. *Impact:* In the above example, not all developers may share the same enthusiasm for test driven development. If it initially results in a negative experience, then they would not prefer to use it again. Apart from this example, the negativity bias leads to decision-makers becoming more risk averse. If an option feels risky, then the decision will be made to avoid it irrespective of whether it could yield a good return with a relatively low factor.	
Debiasing techniques	
List down the facts and try to focus on the positive aspects. Gain a positive perspective by engaging in positive discussion with people who have had a good experience in using the technology.	
Related biases	
Optimism Bias (The tendency to be over-optimistic, overestimating favorable and pleasing outcomes.)	

Table 3.39 Negativity Bias

3.9.3.13 Test Bias

In the software world, different teams adopt different philosophies. Testing is an important phase in the software development life cycle. However, it is not always possible to test all possible use cases is using a software. In case there are no abnormalities when executing such use cases, it leads to an unrealistic confidence that the use case has been successfully implemented without the need for any quantitative results from testing.

Test bias	
Definitions Block	
Some aspects and outcomes of choice cannot be tested, leading to unrealistic confidence in judgement.	
OODA Class: Decide Phase	OODA Subclass: Strategy
Reasoning for classification: There are plenty of testing frameworks and the concept of test driven development is followed by large number of teams. Establishing a sound test strategy ensures a stable code base at all times and reduces failures.	
Examples and impact on architecture design decisions	
Example: Automated frontend testing: Achieving hundred percent code coverage is generally tricky. It is hard to test all aspect of frontend code using automated test frameworks. Realistic expectations have to be set, and it is hard to measure how much testing is realistic. Testing complex corner cases are sometimes avoided if the user interface looks right and works as expected in most cases. *Impact:* In the above use case, it leads to an unrealistic confidence on the test code. It would be harder to detect failures during future deployments as the tests would pass even when there could be a potential error.	
Debiasing techniques	
Identify a testing strategy and establish a testing philosophy. Use test driven development where possible. Always try to keep the tests up to date. Document cases which are hard to test as these could indicate areas of failure in the future.	

Table 3.40 Test Bias

3.9.3.14 Hyperbolic Discounting

Strategies are devised for long-term and short-term. As in any field, there is high pressure to deliver software in short time frames. In such scenarios, software architects tend to design the architectures which can be implemented quickly for immediate payoffs. However, it can have negative impact in the long term as it may not be able to handle all future use cases.

Hyperbolic Discounting	
Definitions Block	
Discounting is the tendency for people to have a stronger preference for more immediate payoffs relative to later payoffs. Hyperbolic discounting leads to choices that are inconsistent over time.	
OODA Class: Decide Phase	OODA Subclass: Strategy
Reasoning for classification: Hyperbolic discounting is based on the principle of least effort (choosing the path of least resistance). In time boxed environments, strategic decisions are generally aimed at making a decision which results in an immediate pay off.	
Examples and impact on architecture design decisions	
Example: CMS versus hand coding websites: CMSs such as Wordpress, Drupal etc. provide a speedy way of implementing websites. Coding websites using websites on the other hand requires considerably more time but offers benefits in terms of being more customizable. *Impact:* A website can be developed in relatively less time using a CMS. However, development of future features will not be feasible if they do not require a high degree of customization.	
Debiasing techniques	
Be mindful of future requirements when making a decision. Verify it as thoroughly as possible. If the option which results in quick pay offs is able to handle more complex requirements in the future, then that could be a viable option. Otherwise, consider other options.	
Related biases	
Inconsistency	

Table 3.41 Hyperbolic Discounting

3.9.3.15 Inconsistency

Software architects are essentially decision-makers. Decision-making involves varying degree of judgments. Inconsistency bias comes into picture when software architects judge similar situations differently and apply different strategies by designing the architecture differently for each case.

Inconsistency	
Definitions Block	
Often a consistent judgement strategy is not applied to an identical repetitive set of cases.	
OODA Class: Decide Phase	OODA Subclass: Strategy
Reasoning for classification: Inconsistency arises when different strategies which are incoherent with one another are applied across projects which share some kind of dependency.	
Examples and impact on architecture design decisions	
Example: Using multiple testing frameworks: Consider the scenario where two teams within an organization are setting up a test framework for frontend projects. Two popular choices would be Jasmine and Jest based on recent trends. If each team decides to go with a different choice of testing framework, then it creates an inconsistency in the technology stack being used within the organization. *Impact:* The direct consequence with an inconsistent strategy is difficulties in integration, especially within a project which has a lot of different components. In case of the above example, a decision may be taken in the future to consolidate the technology stack. All teams could be requested to use a single framework to leverage some common components across projects. In this case, teams which are not using the selected framework would have to migrate to the chosen one increasing development overhead.	
Debiasing techniques	
Brainstorm the different alternatives and freeze on an option. Create a rule set defining which technologies have to be used for which scenarios for reference when setting up future projects.	

Related biases
Hyperbolic discounting

Table 3.42 Inconsistency

3.9.4 Cognitive Biases related to the Act Phase

3.9.4.1 Misinformation Effect

The misinformation effect makes the software architect biased towards a decision which was made previously irrespective of whether is yielded positive results or not.

Misinformation effect	
Definitions Block	
Memory becoming less accurate because of interference from *post-event information.*	
OODA Class: Act Phase	OODA Subclass: N/A
Reasoning for classification: The misinformation effect comes into play after a decision has been made and acted upon. The new information generated from act phase overwrites the information gathered during the observe phase.	
Examples and impact on architecture design decisions	
Example: AWS Cloudfront vs Akamai: Both are viable options to setup a CDN. If Akamai was being used previously, then future use projects would continue to use Akamai as the CDN because it works as expected. *Impact:* While both solutions are good choices, it is easier to find people with AWS expertise than with expertise with Akamai. The misinformation effect influences the software architect towards being biased to Akamai since the choice was made before. Cloudfront may not be considered even it has the potential to be a better solution.	
Debiasing techniques	
Documenting critical information and keeping in touch with it regularly would help in keeping memories more accurate. Moreover, verify if the information and assumptions made in the ‘pre-event’ phase corresponds with the ‘post-event’ information. Update the document in case of any mismatch for future reference.	

Related biases
Post-purchase rationalization

Table 3.43 Misinformation Effect

3.9.4.2 Post-purchase Rationalization

Post-purchase rationalization is similar to misinformation effect. It makes the software architect biased towards a decision which was made previously thereby clouding the future judgements.

Post-purchase Rationalization	
Definitions Block	
The tendency to persuade oneself through rational argument that a purchase was good value.	
OODA Class: Act Phase	OODA Subclass: N/A
Reasoning for classification: The effects of post-purchase rationalization is felt once a decision for a purchase has been made and has been acted upon. The purchase is justified irrespective of whether it was good or bad.	
Examples and impact on architecture design decisions	
Example: Purchase of third-party solutions: Companies often spend large amounts of money in purchasing readymade solutions from third parties. This is especially true in cases when the required skill set is lacking within the organization. *Impact:* Following the purchase of such solutions, the post-purchase rationalization effect would result in the continued application of the selected technology even in cases when it could be a poor choice. The point of realization might come a bit too late until the point of a resounding failure. It would then require vast efforts in large scale migrations from the old technology to a new one. Post-purchase rationalization also makes it harder to learn from new alternatives down the line.	
Debiasing techniques	
If using third party solutions, ensure that support is being provided for a long term along with regular updates. Set checkpoints to regularly verify if the existing solution caters to existing requirements as well as new ones.	

Related biases
Misinformation Effect, Choice-supportive bias (The tendency to remember one's choices as better than they actually were).

Table 3.44 Post-purchase Rationalization

4 Results and Feedback

4 Results and Feedback

4.1 Reflection on the Research Questions

The thesis presented three research questions in the first chapter. The questions revolve around three concepts : DMMs, OODA loop and cognitive biases. The sections summarize the thesis contribution and present the results of the research.

RQ 1 : Which decision-making models are relevant in the context of making software architecture design decisions?

The first question focuses on DMMs. Two approaches towards decision-making were explored during the course of the thesis – normative and behavioral. Several decision-making models falling under both approaches were presented in the Chapter 2 covering related works.

Under the normative approach, three models were examined initially – RE Model, Brunwicks's Lens Model and the Cynefin Framework. From the three models, the RE model was selected for further investigation mainly because the process of decision-making could be clearly represented as individual steps. The Brunswick's Lens model was not considered due to its statistical nature which makes it hard to represent as a sequence of steps. Additionally, it requires the actual decision to be compared with an "ideal" decision which was out of the scope of the thesis. The second model which was not investigated further was the Cynefin Framework as is designed for leaders and policy-makers and was not a suitable model for software architects.

With respect to the behavioral approach, three models were of interest – Incrementalism model, the RPD model and the BR model. The RPD model and BR model, as with the RE model, were selected because they could be represented clearly as sequence of steps. The Incrementalism model was discarded as its focus is on risk mitigation.

Thus, three models of decision-making are relevant for software architects in making architectural decisions as the steps of decision-making could be clearly presented. They are the Rational Economic Model, Recognition-Primed Decision Model and the Bounded Rational Model.

RQ 2 : What is the relationship between the decision-making models and the OODA loop?

To answer the second question, a relationship had to be established between the selected DMMs and the OODA loop decision cycle. This was done by first explaining the concept of OODA Loop and its relevance from the context of making SADD in chapter 4. Afterwards, a relationship was established by intersecting the four phases of the OODA Loop and the three DMMs through the matrix skeleton as shown in figure 1. The individual steps of every DMM is mapped to one of the four phases of the OODA Loop to clearly visualize the relationship in the form of process models as seen in figures 4.1, 4.2 and 4.3.

RQ 3 : Which cognitive biases influence software architects when designing architectures?

Software architects use various heuristics during decision-making. The third research question focusses on bringing to light the different cognitive biases which influence the heuristics used by software architects. To answer this question, cognitive biases from different sources were gathered and over two hundred of them are documented in Appendix 1. Some of them are repetitive in nature, but it is documented nonetheless for the sake of completeness and to reduce publication bias. From amongst these, thirty-three of them were deemed to be relevant from the context of making SADD. However, the list was still too large to be read at one go and is overwhelming at the first glance. To break down the information and to justify the selection of these thirty-three cognitive biases, a two-level cognitive bias classification was made. The four phases of the OODA loop were used to create the first level of classification. This classification was fairly simple as the relevant cognitive biases were classified under one or more phases of the OODA loop. The second level of classification was made to further modularize the long list of cognitive biases under each phase. The labelling of the second-level of classification was done to make the readers (software architects) gain an intuitive understanding of the cognitive biases. Finally, the cognitive bias catalogue in section 4.8 can be used to gain detailed knowledge about all the thirty-three biases from the context of making SADD.

4.2 Using the Thesis Artifacts

Two artifacts were generated during the course of the thesis. The first one is the formalized decision-making process models and the second one is the bias catalogue. The intended target group is software architecture community or any decision-maker involved in making SADD. The next two sections describe the way in which the thesis artifacts can be used by the target group. The actors, however, are free to use the artifacts in any other way as well.

4.2.1 Understanding the DMMs

Software architects often make decisions without being conscious of how the decision-making takes place. Thus, the first step for decision-makers is to explicitly understand what happens during the decision-making process. To understand this, it is best to start with the OODA loop decision cycle as it is one of the most popular decision-making tools used successfully in many fields. This forms the basis for the next step which is to understand the three DMMs. The three DMMs explored in the thesis describe each process of decision-making explicitly in the form of process diagrams. Learning the relationship between the OODA loop and the DMMs aids in makes the information explicitly available in the memory of the decision-maker. This brings the *availability bias* into play making the decision-maker aware of the cognitive biases influencing their decision-making process.

4.2.2 Using the Cognitive Bias Catalogue

The final step is to understand how to use the cognitive bias catalogue. On a first look, the catalogue is extensive and is difficult for most readers to completely read it in one go. The two-level classification is to help deal with the information overload and to reduce the overwhelming feeling induced from it. The classification helps in intuitively understanding which types of cognitive biases come into play during the different stages of decision making. It is then up to the decision-maker to realize which stage of decision-making is in progress and then read about the cognitive biases which influence that particular stage. The availability bias comes into picture due to the availability of the information in memory making the decision-maker conscious of them. This helps the decision-maker to be aware of the possible influences that those cognitive biases could have and try and reduce any negative impact.

4.3 Evaluation through Expert Feedback

To conduct the evaluation of the thesis, an expert feedback methodology was adopted. A simple website with four pages was made to present the findings of the research. The first page was the landing page and consisted of the abstract of the thesis as well as a short description on how to use the thesis artifacts. The second page consisted of information about the two decision-making approaches, OODA loop and the three DMMs. The third page consisted of the bias catalogue for the reader to learn about the cognitive biases. The last page was to collect feedback and had a simple google form embedded in it. The feedback form is presented in Appendix 2. The screenshots from the website are presented from Figure 4.1 through to 4.5.

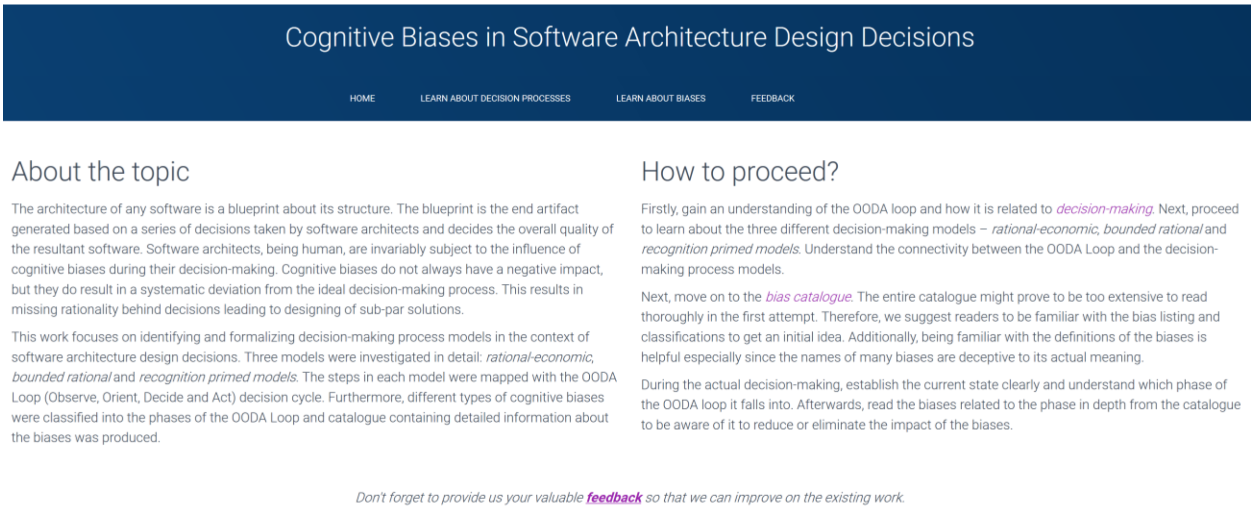


Figure 4.1 Landing page

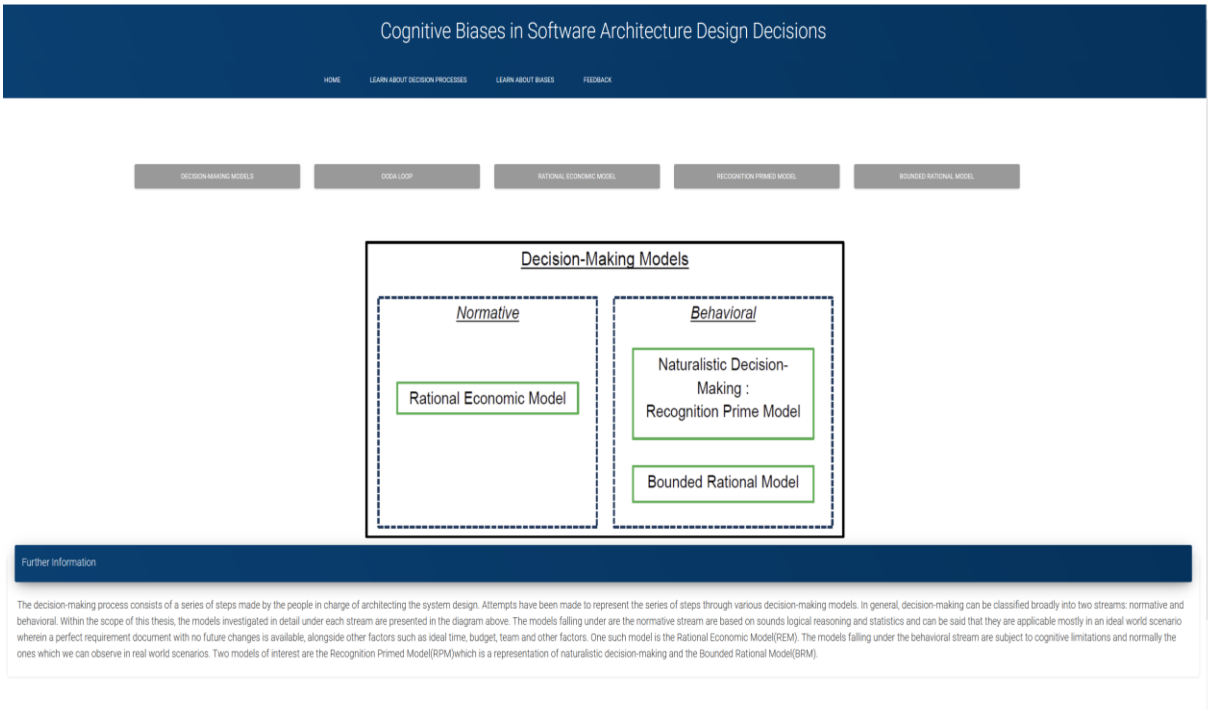


Figure 4.2 Decision-making models with brief descriptions

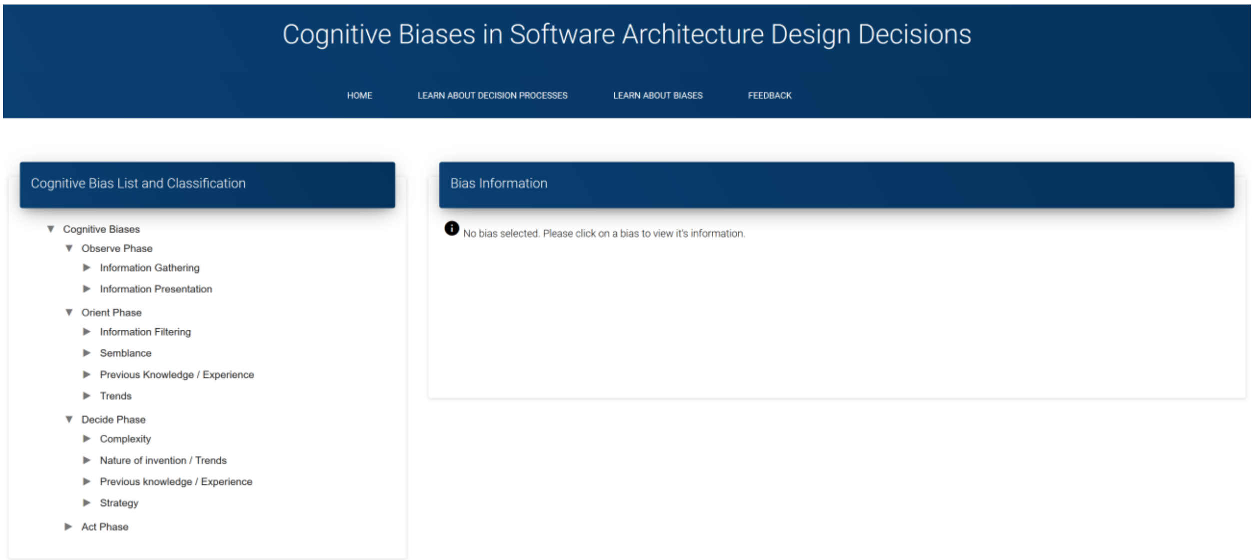


Figure 4.3 Bias Catalogue Page (when no bias is selected)

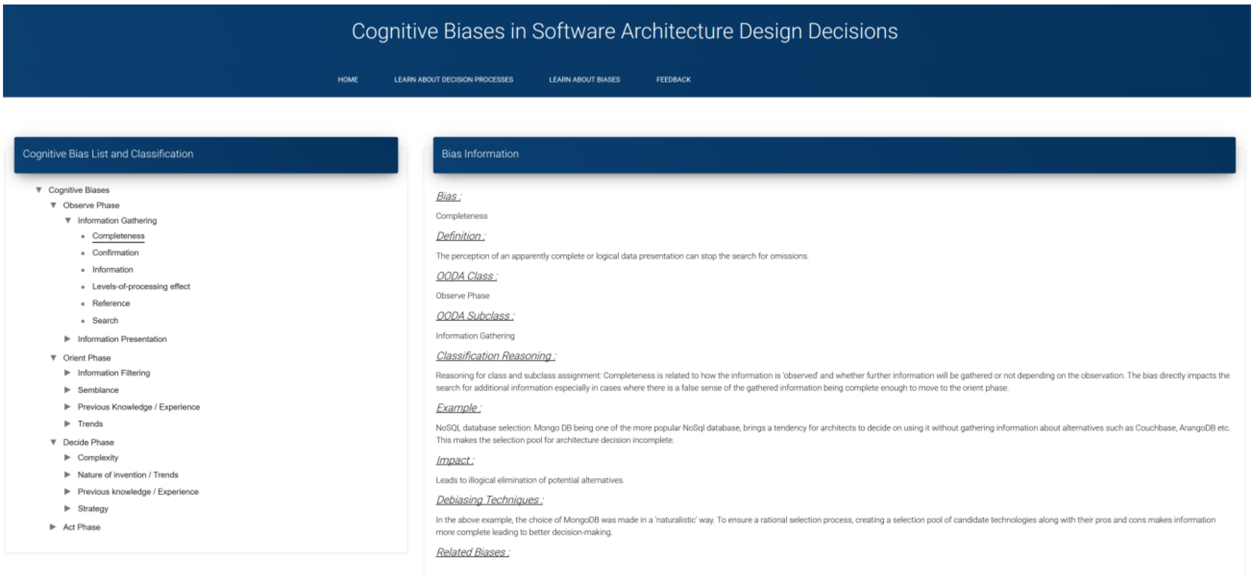


Figure 4.4 Bias Catalogue Page (when a bias is selected)

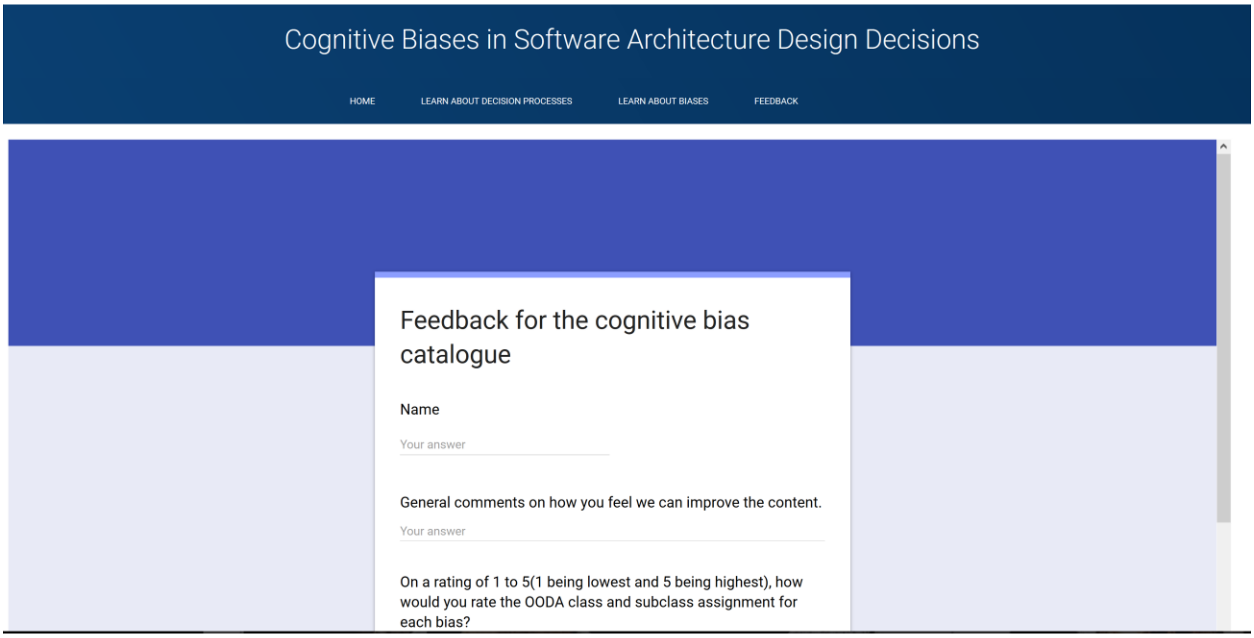


Figure 4.5 Feedback page with Google form

Rigorous evaluation was not in the scope of the thesis and limited time was spent on this. Around fifteen people were identified to gather information from through personal contacts and in the end seven people provided feedback. The final participant list consisted of seven people working in capacity as either software architects, lead developers or as product owners. A statistical summary of responses for each question in the feedback form is presented from figure 4.6 through to figure 4.8.

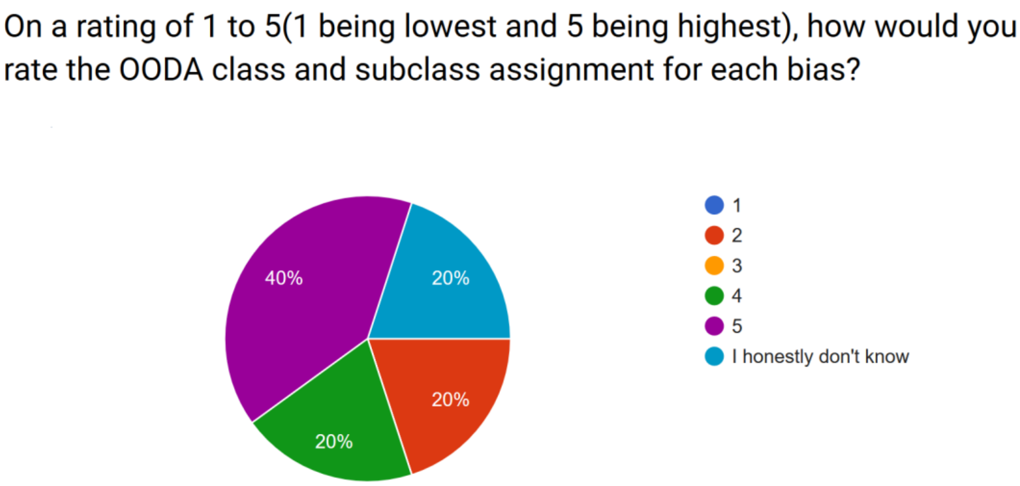


Figure 4.6 First question regarding the two-level classification

The first question yielded a moderate response. On contacting the people who did not rate it highly, it was found that they felt that they could not judge the two-level classification. This was because no such classification existed to compare with in this context.

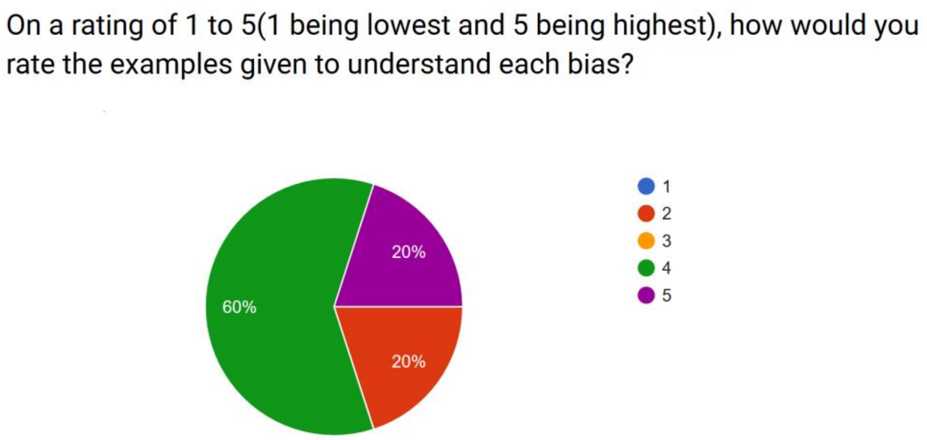


Figure 4.7 Second question regarding the examples presented in the cognitive bias catalogue

The second question regarding the quality of the example resulted in a majority of the participants liking them. The person who gave the low rating reasoned there was an overload of information and that the manner in which the information was presented was lacking in interactivity. A suggestion was made to create a mobile app to potentially gamify the learning process.

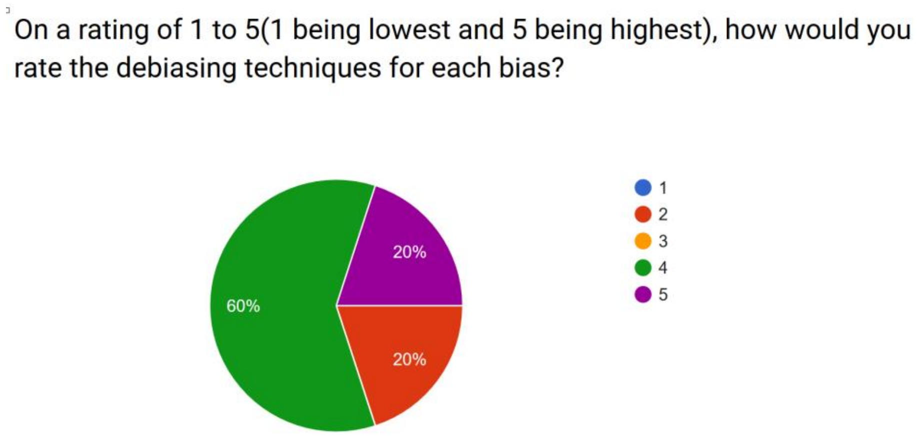


Figure 4.8 Third question regarding the debiasing techniques presented in the cognitive bias catalogue

The third and the final question asked the participants to rate the debiasing techniques. The debiasing techniques mentioned was also liked by the majority of them. The reason for one participant rating it low was again due to an overload of information and due to a lack of interactive learning process in the website.

Here are some quotes gathered during the feedback stage.

- “We are biased because we take a look at what worked in the past and trust our future decisions based on it..”
- “Content is good and was worth reading.“
- “Lots of biases and too much information. Reading all of it was intensive.”
- “It would be nice 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..”
- “The biases and classification feel genuine. The question next is how to rectify them.“

5 Future Work and Conclusion

5 Future Work and Conclusion

5.1 Future Work

5.1.1 Cognitive Bias Detection Engine

Cognitive biases is increasingly becoming a popular topic in the field of software engineering. With the emergence of artificial intelligence and automated decision support systems, it is imperative to evaluate systems for biases. In this direction, the cognitive bias catalogue can be used to implement a bias detection engine to improve existing decision support systems.

5.1.2 Improving the cognitive bias classification and catalogue

The cognitive bias catalogue was an attempt at aggregating possible cognitive biases which influence software architects when designing architectures. There will always be new types of cognitive biases relevant in this context which could be discovered in the future and the two-level classification will need to be updated to accommodate the new biases.

Another area of potential improvement in the future is the content in cognitive bias catalogue. The information in the catalogue is intended for software architects to gain a thorough understanding of each cognitive bias. Definitions from additional sources can included to provide additional perspective to reduce *publication bias.* The examples were kept simple so that software architects with varying amounts of experience and domain expertise can relate to it. More examples can be added in the future covering detailed scenarios of varying complexities which could be tailored for specific domains. Exploring debiasing techniques was not in the scope of the thesis, but was included for the purpose of making the catalogue complete. The techniques to debias are basic in nature and further research will have to be conducted in the future to find specific ways of debiasing each cognitive bias. This was also mentioned in one of the feedbacks.

5.1.3 Interactive ways of information presenting

Another area for future work is in the presentation of the bias catalogue. The website through which we presented the information was static and lacked any interaction with the readers. Gamification of the learning process was suggested in one of the feedbacks. Mobile apps for learning is another way for continuous learning. It was also discussed that it can be incorporated in corporate trainings to educate software architects about the DMMs and cognitive biases.

5.2 Conclusion

The thesis revolves around exploring three concepts – the OODA loop decision-cycle, decision-making models and cognitive biases. The concepts are explored from the point of view of designing software architecture as most software architects have their own way of decision-making and is often unstructured. This makes it difficult to establish a generic pattern of decision-making and was the reason behind undertaking this research. To establish this generic pattern, the OODA loop was explored in detail as it is one of the most popular decision-making tools used by successful decision-makers in different fields. A clear relationship is established between the OODA loop and the DMMs generally used in making SADD. Through this, a generic pattern of structured decision-making which can be adopted by software architects is established in the thesis. Additionally, the thirty-three cognitive biases presented as part of the cognitive bias catalogue aids in increasing the awareness of software architects on the potential impact of cognitive bases when making SADD.

III Bibliography

[1] “A Cognitive bias cheat sheet – Better Humans”. Better Humans, 2016-09-01.

[2] Arnott, D. (2006), Cognitive biases and decision support systems development: a design science approach. Information Systems Journal, 16: 55–78. doi:10.1111/j.1365-2575.2006.00208.x

[3] Arnott, D. (1998). A taxonomy of decision biases. Monash University, School of Information Management and Systems, Caulfield.

[4] A. Jansen and J. Bosch, "Software Architecture as a Set of Architectural Design Decisions," 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05), Pittsburgh, PA, USA, 2005, pp. 109-120.

[5] Barros, Gustavo. (2010). Herbert A. Simon and the concept of rationality: boundaries and procedures. *Brazilian Journal of Political Economy*, *30*(3), 455-472.

[6] Benson, Buster, "Cognitive bias cheat sheet", *Better Humans*, Sep 1.2016.

[7] Burge, Janet E. Rationale-Based Software Engineering. Springer, 2010.

[8] Crowder, M. (2013). Decision-making: Two schools of thought or one? In V. Sheppard (Ed), Proceedings of the Salford Postgraduate Annual Research Conference (SPARC) 2012 (pp 47-63). Salford: University of Salford.

[9] C. Zannier, F. Maurer, "Social Factors Relevant to Capturing Design Decisions", In Proceedings of the Second Workshop on Sharing and Reusing Architectural Knowledge (SHARK-ADI07), 2007.

[10] David G. Ullman. “OO-OO-OO!” the sound of a broken OODA loop. Crosstalk,April 2007.

[11] Davide Falessi, Giovanni Canton, Rick Kazman, Philippe Kruchten, Decision-making techniques for software architecture design: A comparative survey, ACM Computing Surveys (CSUR), v.43 n.4, p.1-28, October 2011.

[12] Durdik, Zoya. Architectural Design Decision Documentation through Reuse of Design Patterns. KIT Scientific Publishing, 2016.

[13] G. Pedraza-García, H. Astudillo and D. Correal, "Modeling Software Architecture Process with a Decision-Making Approach," 2014 33rd International Conference of the Chilean Computer Science Society (SCCC), Talca, 2014, pp. 1-6.

[14] Hammond, Grant Tedrick. *The Mind of War: John Boyd and American Security*. Smithsonian Books, 2012.

[15] Hans van Vliet , Antony Tang, Decision making in software architecture, Journal of Systems and Software, v.117 n.C, p.638-644, July 2016

[16] Kahneman, D; Tversky, (1972). “Subjective probability: A judgement of representativeness”. Cognitibe Psychology/

[17] “List of Cognitive Biases”, n.d.

[18] McKay, Brett & Kate. “OODA Loop: A Comprehensive Guide.” *The Art of Manliness*, 8 Jan. 2018, www.artofmanliness.com/2014/09/15/ooda-loop/.

[19] Munier, B., Selten, R., Bouyssou, D. et al. Marketing Letters (1999) 10: 233.

[20] Osinga, Frans P. B. *Science, Strategy and War: the Strategic Theory of John Boyd*. Routledge, 2007.

[21] Stephen T. Hassard, Ann Blandford, Anna L. Cox, Analogies in design decision-making, Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology, p.140-148, September 01-05, 2009, Cambridge, United Kingdom

[22] S. Bhukya and S. Pabboju, "Software architecture central concern, key decisions," 2016 International Conference on Electrical,k Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 2451-2457.

[23] S. Dasanayake, J. Markkula, S. Aaramaa and M. Oivo, "Software Architecture Decision-Making Practices and Challenges: An Industrial Case Study," 2015 24th Australasian Software Engineering Conference, Adelaide, SA, 2015, pp. 88-97.

[24] Tang, Antony and Man F. Lau. 2014. Software architecture review by association. *J. Syst. Softw.* 88, C (February 2014), 87-101.

[25] Tang, Antony, M. Razavian, B. Paech and T. M. Hesse, "Human Aspects in Software Architecture Decision Making: A Literature Review," *2017 IEEE International Conference on Software Architecture (ICSA)*, Gothenburg, 2017, pp. 107-116.

[26] Tang, Antony. "Software designers, are you biased?" Proceedings of the 6th International Workshop on Sharing and Reusing Architectural Knowledge. ACM, 2011.

[27] Tang. Antony, M. Razavian, B. Paech and T. M. Hesse, "Human Aspects in Software Architecture Decision Making: A Literature Review," 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 2017, pp. 107-116.

[28] T. D. LaToza, E. Shabani and A. van der Hoek, "A study of architectural decision practices," 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), San Francisco, CA, 2013, pp. 77-80.

[29] van der Ven J.S., Jansen A.G.J., Nijhuis J.A.G., Bosch J. (2006) Design Decisions: The Bridge between Rationale and Architecture. In: Dutoit A.H., McCall R., Mistrík I., Paech B. (eds) Rationale Management in Software Engineering. Springer, Berlin, Heidelberg

[30] Zalewski Andrzej, Ratkowski Andrzej, On Cognitive Biases in Architecture Decision Making, 2017.

IV Appendix 1 – List of Cognitive Biases [2][17]

Name	Description
Actor-observer bias	The tendency for explanations of other individuals' behaviors to overemphasize the influence of their personality and underemphasize the influence of their situation (see also Fundamental attribution error), and for explanations of one's own behaviors to do the opposite (that is, to overemphasize the influence of our situation and underemphasize the influence of our own personality)
Ambiguity effect	The tendency to avoid options for which missing information makes the probability seem "unknown"
Anchoring and adjustment	Adjustments from an initial position are usually insufficient
Anchoring or focalism	The tendency to rely too heavily, or "anchor", on one trait or piece of information when making decisions (usually the first piece of information acquired on that subject)
Anthropocentric thinking	The tendency to use human analogies as a basis for reasoning about other, less familiar, biological phenomena
Anthropomorphism or personification	The tendency to characterize animals, objects, and abstract concepts as possessing human-like traits, emotions, and intentions
Attentional bias	The tendency of our perception to be affected by our recurring thoughts
Attenuation	A decision-making situation can be simplified by ignoring or significantly discounting the level of uncertainty
Authority bias	The tendency to attribute greater accuracy to the opinion of an authority figure (unrelated to its content) and be more influenced by that opinion
Automation bias	The tendency to depend excessively on automated systems which can lead to erroneous automated information overriding correct decisions
Availability cascade	A self-reinforcing process in which a collective belief gains more and more plausibility through its increasing repetition in public discourse (or "repeat something long enough and it will become true")

Availability heuristic	The tendency to overestimate the likelihood of events with greater "availability" in memory, which can be influenced by how recent the memories are or how unusual or emotionally charged they may be
Backfire effect	The reaction to disconfirming evidence by strengthening one's previous beliefs. cf. Continued influence effect
Bandwagon effect	The tendency to do (or believe) things because many other people do (or believe) the same. Related to groupthink and herd behavior
Base rate	Base rate data tends to be ignored when other data are available
Base rate fallacy or Base rate neglect	The tendency to ignore base rate information (generic, general information) and focus on specific information (information only pertaining to a certain case)
Belief bias	An effect where someone's evaluation of the logical strength of an argument is biased by the believability of the conclusion
Ben Franklin effect	A person who has performed a favor for someone is more likely to do another favor for that person than they would be if they had received a favor from that person.
Berkson's paradox	The tendency to misinterpret statistical experiments involving conditional probabilities.
Bias blind spot	The tendency to see oneself as less biased than other people, or to be able to identify more cognitive biases in others than in oneself
Bizarreness effect	Bizarre material is better remembered than common material.
Chance	A sequence of random events can be mistaken for an essential characteristic of a process
Change bias	After an investment of effort in producing change, remembering one's past performance as more difficult than it actually was
Cheerleader effect	The tendency for people to appear more attractive in a group than in isolation
Childhood amnesia	The retention of few memories from before the age of four

Choice-supportive bias	The tendency to remember one's choices as better than they actually were
Choice-supportive bias	In a self-justifying manner retroactively ascribing one's choices to be more informed than they were when they were made.
Clustering illusion	The tendency to overestimate the importance of small runs, streaks, or clusters in large samples of random data (that is, seeing phantom patterns)
Completeness	The perception of an apparently complete or logical data presentation can stop the search for omissions
Complexity	Time pressure, information overload and other environmental factors can increase the perceived complexity of a task
Confirmation	Often decision-makers seek confirmatory evidence and do not search for disconfirming information
Confirmation bias	The tendency to search for, interpret, focus on and remember information in a way that confirms one's preconceptions
Congruence bias	The tendency to test hypotheses exclusively through direct testing, instead of testing possible alternative hypotheses
Conjunction	Probability is often overestimated in compound conjunctive problems
Conjunction fallacy	The tendency to assume that specific conditions are more probable than general ones
Conservatism	Often estimates are not revised appropriately on the receipt of significant new data
Conservatism (belief revision)	The tendency to revise one's belief insufficiently when presented with new evidence
Conservatism or Regressive bias	Tendency to remember high values and high likelihoods/probabilities/frequencies as lower than they actually were and low ones as higher than they actually were. Based on the evidence, memories are not extreme enough

Consistency bias	Incorrectly remembering one's past attitudes and behaviour as resembling present attitudes and behaviour
Context effect	That cognition and memory are dependent on context, such that out-of-context memories are more difficult to retrieve than in-context memories (e.g., recall time and accuracy for a work-related memory will be lower at home, and vice versa)
Continued influence effect	The tendency to believe previously learned misinformation even after it has been corrected. Misinformation can still influence inferences one generates after a correction has occurred. cf. Backfire effect
Contrast effect	The enhancement or reduction of a certain stimulus' perception when compared with a recently observed, contrasting object
Control	A poor decision may lead to a good outcome, inducing a false feeling of control over the judgement situation
Correlation	The probability of two events occurring together can be overestimated if they have co-occurred in the past
Courtesy bias	The tendency to give an opinion that is more socially correct than one's true opinion, so as to avoid offending anyone
Cross-race effect	The tendency for people of one race to have difficulty identifying members of a race other than their own
Cryptomnesia	A form of misattribution where a memory is mistaken for imagination, because there is no subjective experience of it being a memory
Curse of knowledge	When better-informed people find it extremely difficult to think about problems from the perspective of lesser-informed people
Declinism	The belief that a society or institution is tending towards decline. Particularly, it is the predisposition to view the past favourably (rosy retrospection) and future negatively
Decoy effect	Preferences for either option A or B change in favor of option B when option C is presented, which is similar to option B but in no way better.

Defensive attribution hypothesis	Attributing more blame to a harm-doer as the outcome becomes more severe or as personal or situational similarity to the victim increases.
Denomination effect	The tendency to spend more money when it is denominated in small amounts (e.g., coins) rather than large amounts (e.g., bills)
Desire	The probability of desired outcomes may be inaccurately assessed as being greater
Disjunction	Probability is often underestimated in compound disjunctive problems
Disposition effect	The tendency to sell an asset that has accumulated in value and resist selling an asset that has declined in value.
Distinction bias	The tendency to view two options as more dissimilar when evaluating them simultaneously than when evaluating them separately
Dunning–Kruger effect	The tendency for unskilled individuals to overestimate their own ability and the tendency for experts to underestimate their own ability
Duration neglect	The neglect of the duration of an episode in determining its value
Egocentric bias	Occurs when people claim more responsibility for themselves for the results of a joint action than an outside observer would credit them with.
Egocentric bias	Recalling the past in a self-serving manner, e.g., remembering one's exam grades as being better than they were, or remembering a caught fish as bigger than it really was.
Empathy gap	The tendency to underestimate the influence or strength of feelings, in either oneself or others.
Endowment effect	The tendency for people to demand much more to give up an object than they would be willing to pay to acquire it
Escalation	Often decision-makers commit to follow or escalate a previous unsatisfactory course of action
Exaggerated expectation	Based on the estimates, real-world evidence turns out to be less extreme than our expectations (conditionally inverse of the conservatism bias

Experimenter's or expectation bias	The tendency for experimenters to believe, certify, and publish data that agree with their expectations for the outcome of an experiment, and to disbelieve, discard, or downgrade the corresponding weightings for data that appear to conflict with those expectations
Extrinsic incentives bias	An exception to the fundamental attribution error, when people view others as having (situational) extrinsic motivations and (dispositional) intrinsic motivations for oneself
Fading affect bias	A bias in which the emotion associated with unpleasant memories fades more quickly than the emotion associated with positive events
False consensus effect	The tendency for people to overestimate the degree to which others agree with them
False memory	A form of misattribution where imagination is mistaken for a memory.
Focusing effect	The tendency to place too much importance on one aspect of an event
Forer effect (aka Barnum effect)	The tendency to give high accuracy ratings to descriptions of their personality that supposedly are tailored specifically for them, but are in fact vague and general enough to apply to a wide range of people. For example, horoscopes.
Forer effect or Barnum effect	The observation that individuals will give high accuracy ratings to descriptions of their personality that supposedly are tailored specifically for them, but are in fact vague and general enough to apply to a wide range of people. This effect can provide a partial explanation for the widespread acceptance of some beliefs and practices, such as astrology, fortune telling, graphology, and some types of personality tests.
Framing	Events framed as either losses or gains may be evaluated differently
Framing effect	Drawing different conclusions from the same information, depending on how that information is presented
Frequency illusion	The illusion in which a word, a name, or other thing that has recently come to one's attention suddenly seems to appear with improbable frequency shortly afterwards (not to be confused with the recency

	illusion or selection bias).This illusion may explain some examples of the Baader-Meinhof phenomenon, when someone repeatedly notices a newly learned word or phrase shortly after learning it
Functional fixedness	Limits a person to using an object only in the way it is traditionally used
Fundamental attribution error	The tendency for people to over-emphasize personality-based explanations for behaviors observed in others while under-emphasizing the role and power of situational influences on the same behavior (see also actor-observer bias, group attribution error, positivity effect, and negativity effect)
Gambler's fallacy	The tendency to think that future probabilities are altered by past events, when in reality they are unchanged. The fallacy arises from an erroneous conceptualization of the law of large numbers. For example, "I've flipped heads with this coin five times consecutively, so the chance of tails coming out on the sixth flip is much greater than heads."
Generation effect (Self-generation effect)	That self-generated information is remembered best. For instance, people are better able to recall memories of statements that they have generated than similar statements generated by others.
Google effect	The tendency to forget information that can be found readily online by using Internet search engines.
Group attribution error	The biased belief that the characteristics of an individual group member are reflective of the group as a whole or the tendency to assume that group decision outcomes reflect the preferences of group members, even when information is available that clearly suggests otherwise.
Habit	An alternative may be chosen only because it was used before
Halo effect	The tendency for a person's positive or negative traits to "spill over" from one personality area to another in others' perceptions of them (see also physical attractiveness stereotype)
Hard–easy effect	Based on a specific level of task difficulty, the confidence in judgments is too conservative and not extreme enough

Hindsight	In retrospect, the degree to which an event could have been predicted is often overestimated
Hindsight bias	Sometimes called the "I-knew-it-all-along" effect, the tendency to see past events as being predictable at the time those events happened.
Hindsight bias	The inclination to see past events as being more predictable than they actually were; also called the "I-knew-it-all-along" effect.
Hostile attribution bias	The "hostile attribution bias" is the tendency to interpret others' behaviors as having hostile intent, even when the behavior is ambiguous or benign.
Hot-hand fallacy	The "hot-hand fallacy" (also known as the "hot hand phenomenon" or "hot hand") is the fallacious belief that a person who has experienced success with a random event has a greater chance of further success in additional attempts.
Humor effect	That humorous items are more easily remembered than non-humorous ones, which might be explained by the distinctiveness of humor, the increased cognitive processing time to understand the humor, or the emotional arousal caused by the humor
Hyperbolic discounting	Discounting is the tendency for people to have a stronger preference for more immediate payoffs relative to later payoffs. Hyperbolic discounting leads to choices that are inconsistent over time – people make choices today that their future selves would prefer not to have made, despite using the same reasoning. Also known as current moment bias, present-bias, and related to Dynamic inconsistency.
Identifiable victim effect	The tendency to respond more strongly to a single identified person at risk than to a large group of people at risk
IKEA effect	The tendency for people to place a disproportionately high value on objects that they partially assembled themselves, such as furniture from IKEA, regardless of the quality of the end result
Illusion of asymmetric insight	People perceive their knowledge of their peers to surpass their peers' knowledge of them

Illusion of control	The tendency to overestimate one's degree of influence over other external events
Illusion of external agency	When people view self-generated preferences as instead being caused by insightful, effective and benevolent agents
Illusion of transparency	People overestimate others' ability to know them, and they also overestimate their ability to know others.
Illusion of truth effect	That people are more likely to identify as true statements those they have previously heard (even if they cannot consciously remember having heard them), regardless of the actual validity of the statement. In other words, a person is more likely to believe a familiar statement than an unfamiliar one.
Illusion of validity	Belief that our judgments are accurate, especially when available information is consistent or inter-correlated
Illusory correlation	Inaccurately perceiving a relationship between two unrelated events
Illusory correlation	Inaccurately remembering a relationship between two events
Illusory superiority	Overestimating one's desirable qualities, and underestimating undesirable qualities, relative to other people. (Also known as "Lake Wobegon effect", "better-than-average effect", or "superiority bias")
Illusory truth effect	A tendency to believe that a statement is true if it is easier to process, or if it has been stated multiple times, regardless of its actual veracity. These are specific cases of truthiness
Imaginability	An event may be judged more probable if it can be easily imagined
Impact bias	The tendency to overestimate the length or the intensity of the impact of future feeling states
Inconsistency	Often a consistent judgement strategy is not applied to an identical repetitive set of cases
Information bias	The tendency to seek information even when it cannot affect action
Ingroup bias	The tendency for people to give preferential treatment to others they perceive to be members of their own groups.

Insensitivity to sample size	The tendency to under-expect variation in small samples.
Irrational escalation	The phenomenon where people justify increased investment in a decision, based on the cumulative prior investment, despite new evidence suggesting that the decision was probably wrong. Also known as the sunk cost fallacy
Just-world hypothesis	The tendency for people to want to believe that the world is fundamentally just, causing them to rationalize an otherwise inexplicable injustice as deserved by the victim(s)
Lag effect	The phenomenon whereby learning is greater when studying is spread out over time, as opposed to studying the same amount of time in a single session. See also spacing effect
Law of the instrument	An over-reliance on a familiar tool or methods, ignoring or under-valuing alternative approaches. "If all you have is a hammer, everything looks like a nail."
Less-is-better effect	The tendency to prefer a smaller set to a larger set judged separately, but not jointly
Leveling and sharpening	Memory distortions introduced by the loss of details in a recollection over time, often concurrent with sharpening or selective recollection of certain details that take on exaggerated significance in relation to the details or aspects of the experience lost through leveling. Both biases may be reinforced over time, and by repeated recollection or re-telling of a memory
Levels-of-processing effect	That different methods of encoding information into memory have different levels of effectiveness
Linear	Decision-makers are often unable to extrapolate a nonlinear growth process
List-length effect	A smaller percentage of items are remembered in a longer list, but as the length of the list increases, the absolute number of items remembered increases as well. For example, consider a list of 30 items

	("L30") and a list of 100 items ("L100"). An individual may remember 15 items from L30, or 50%, whereas the individual may remember 40 items from L100, or 40%. Although the percent of L30 items remembered (50%) is greater than the percent of L100 (40%), more L100 items (40) are remembered than L30 items (15)
Look-elsewhere effect	An apparently statistically significant observation may have actually arisen by chance because of the size of the parameter space to be searched
Loss aversion	The disutility of giving up an object is greater than the utility associated with acquiring it (see also Sunk cost effects and endowment effect).
Mere exposure effect	The tendency to express undue liking for things merely because of familiarity with them
Misinformation effect	Memory becoming less accurate because of interference from post-event information
Modality effect	That memory recall is higher for the last items of a list when the list items were received via speech than when they were received through writing.
Mode	The mode and mixture of presentation can influence the perceived value of data
Money illusion	The tendency to concentrate on the nominal value (face value) of money rather than its value in terms of purchasing power
Mood-congruent memory bias	The improved recall of information congruent with one's current mood.
Moral credential effect	The tendency of a track record of non-prejudice to increase subsequent prejudice.
Moral luck	The tendency for people to ascribe greater or lesser moral standing based on the outcome of an event.
Naïve cynicism	Expecting more egocentric bias in others than in oneself.

Naïve realism	The belief that we see reality as it really is – objectively and without bias; that the facts are plain for all to see; that rational people will agree with us; and that those who don't are either uninformed, lazy, irrational, or biased.
Negativity bias or Negativity effect	Psychological phenomenon by which humans have a greater recall of unpleasant memories compared with positive memories. (see also actor-observer bias, group attribution error, positivity effect, and negativity effect)
Neglect of probability	The tendency to completely disregard probability when making a decision under uncertainty
Next-in-line effect	That a person in a group has diminished recall for the words of others who spoke immediately before himself, if they take turns speaking
Normalcy bias	The refusal to plan for, or react to, a disaster which has never happened before
Not invented here	Aversion to contact with or use of products, research, standards, or knowledge developed outside a group. Related to IKEA effect
Observer-expectancy effect	When a researcher expects a given result and therefore unconsciously manipulates an experiment or misinterprets data in order to find it (see also subject-expectancy effect)
Omission bias	The tendency to judge harmful actions as worse, or less moral, than equally harmful omissions (inactions)
Optimism bias	The tendency to be over-optimistic, overestimating favorable and pleasing outcomes (see also wishful thinking, valence effect, positive outcome bias)
Order	The first or last item presented may be overweighted in judgement
Ostrich effect	Ignoring an obvious (negative) situation
Outcome bias	The tendency to judge a decision by its eventual outcome instead of based on the quality of the decision at the time it was made

Outgroup homogeneity bias	Individuals see members of their own group as being relatively more varied than members of other groups
Overconfidence	The ability to solve difficult or novel problems is often overestimated
Overconfidence effect	Excessive confidence in one's own answers to questions. For example, for certain types of questions, answers that people rate as "99% certain" turn out to be wrong 40% of the time
Pareidolia	A vague and random stimulus (often an image or sound) is perceived as significant, e.g., seeing images of animals or faces in clouds, the man in the moon, and hearing non-existent hidden messages on records played in reverse
Part-list cueing effect	That being shown some items from a list and later retrieving one item causes it to become harder to retrieve the other items.
Peak-end rule	That people seem to perceive not the sum of an experience but the average of how it was at its peak (e.g., pleasant or unpleasant) and how it ended
Persistence	The unwanted recurrence of memories of a traumatic event
Pessimism bias	The tendency for some people, especially those suffering from depression, to overestimate the likelihood of negative things happening to them
Picture superiority effect	The notion that concepts that are learned by viewing pictures are more easily and frequently recalled than are concepts that are learned by viewing their written word form counterparts
Planning fallacy	The tendency to underestimate task-completion times
Positivity effect (Socioemotional selectivity theory)	That older adults favor positive over negative information in their memories
Post-purchase rationalization	The tendency to persuade oneself through rational argument that a purchase was good value

Primacy effect, recency effect & serial position effect	That items near the end of a sequence are the easiest to recall, followed by the items at the beginning of a sequence; items in the middle are the least likely to be remembered
Processing difficulty effect	That information that takes longer to read and is thought about more (processed with more difficulty) is more easily remembered.
Pro-innovation bias	The tendency to have an excessive optimism towards an invention or innovation's usefulness throughout society, while often failing to identify its limitations and weaknesses.
Projection bias	The tendency to overestimate how much our future selves share one's current preferences, thoughts and values, thus leading to sub-optimal choices.
Pseudocertainty effect	The tendency to make risk-averse choices if the expected outcome is positive, but make risk-seeking choices to avoid negative outcomes.
Reactance	The urge to do the opposite of what someone wants you to do out of a need to resist a perceived attempt to constrain your freedom of choice (see also Reverse psychology).
Reactive devaluation	Devaluing proposals only because they purportedly originated with an adversary.
Recall	An event or class may appear more numerous or frequent if its instances are more easily recalled than other equally probable events
Recency illusion	The illusion that a word or language usage is a recent innovation when it is in fact long-established (see also frequency illusion).
Redundancy	The more redundant and voluminous the data, the more confidence may be expressed in its accuracy and importance
Reference	The establishment of a reference point or anchor can be a random or distorted act
Regression	That events will tend to regress towards the mean on subsequent trials is often not allowed for in judgement

Regressive bias	A certain state of mind wherein high values and high likelihoods are overestimated while low values and low likelihoods are underestimated
Reminiscence bump	The recalling of more personal events from adolescence and early adulthood than personal events from other lifetime periods
Restraint bias	The tendency to overestimate one's ability to show restraint in the face of temptation
Rhyme as reason effect	Rhyming statements are perceived as more truthful. A famous example being used in the O.J Simpson trial with the defense's use of the phrase "If the gloves don't fit, then you must acquit."
Risk compensation / Peltzman effect	The tendency to take greater risks when perceived safety increases
Rosy retrospection	The remembering of the past as having been better than it really was
Rule	The wrong decision rule may be used
Sample	The size of a sample is often ignored in judging its predictive power
Scale	The perceived variability of data can be affected by the scale of the data
Search	An event may seem more frequent because of the effectiveness of the search strategy
Selective perception	The tendency for expectations to affect perception.
Selectivity	Expectation of the nature of an event can bias what information is thought to be relevant
Self-relevance effect	That memories relating to the self are better recalled than similar information relating to others
Self-serving bias	The tendency to claim more responsibility for successes than failures. It may also manifest itself as a tendency for people to evaluate ambiguous information in a way beneficial to their interests (see also group-serving bias)
Semmelweis reflex	The tendency to reject new evidence that contradicts a paradigm

Sexual overperception bias / sexual underperception bias	The tendency to over-/underestimate sexual interest of another person in oneself
Shared information bias	Known as the tendency for group members to spend more time and energy discussing information that all members are already familiar with (i.e., shared information), and less time and energy discussing information that only some members are aware of (i.e., unshared information)
Similarity	The likelihood of an event occurring may be judged by the degree of similarity with the class it is perceived to belong to
Sociability bias of language	The disproportionally higher representation of words related to social interactions, in comparison to words related to physical or mental aspects of behavior, in most languages. This bias attributed to nature of language as a tool facilitating human interactions. When verbal descriptors of human behavior are used as a source of information, sociability bias of such descriptors emerges in factor-analytic studies as a factor related to pro-social behavior (for example, of Extraversion factor in the Big Five personality traits
Social comparison bias	The tendency, when making decisions, to favor potential candidates who don't compete with one's own particular strengths
Social desirability bias	The tendency to over-report socially desirable characteristics or behaviours in oneself and under-report socially undesirable characteristics or behaviour
Source confusion	Confusing episodic memories with other information, creating distorted memories
Spacing effect	That information is better recalled if exposure to it is repeated over a long span of time rather than a short one

Spotlight effect	The tendency to overestimate the amount that other people notice your appearance or behavior
Status quo bias	The tendency to like things to stay relatively the same (see also loss aversion, endowment effect, and system justification)
Stereotypical bias	Memory distorted towards stereotypes (e.g., racial or gender), e.g., "black-sounding" names being misremembered as names of criminals
Stereotyping	Expecting a member of a group to have certain characteristics without having actual information about that individual
Subadditivity effect	The tendency to judge probability of the whole to be less than the probabilities of the parts
Subjective validation	Perception that something is true if a subject's belief demands it to be true. Also assigns perceived connections between coincidences.
Subset	A conjunction or subset is often judged more probable than its set
Success	Often failure is associated with poor luck, and success with the abilities of the decision-maker
Suffix effect	Diminishment of the recency effect because a sound item is appended to the list that the subject is not required to recall
Suggestibility	A form of misattribution where ideas suggested by a questioner are mistaken for memory
Surrogation	Losing sight of the strategic construct that a measure is intended to represent, and subsequently acting as though the measure is the construct of interest
Survivorship bias	Concentrating on the people or things that "survived" some process and inadvertently overlooking those that didn't because of their lack of visibility
System justification	The tendency to defend and bolster the status quo. Existing social, economic, and political arrangements tend to be preferred, and alternatives disparaged, sometimes even at the expense of individual and collective self-interest. (See also status quo bias.)

Telescoping effect	The tendency to displace recent events backward in time and remote events forward in time, so that recent events appear more remote, and remote events, more recent
Test	Some aspects and outcomes of choice cannot be tested, leading to unrealistic confidence in judgement
Testimony	The inability to recall details of an event may lead to seemingly logical reconstructions that may be inaccurate
Testing effect	The fact that you more easily remember information you have read by rewriting it instead of rereading it
Third-person effect	Belief that mass communicated media messages have a greater effect on others than on themselves
Time-saving bias	Underestimations of the time that could be saved (or lost) when increasing (or decreasing) from a relatively low speed and overestimations of the time that could be saved (or lost) when increasing (or decreasing) from a relatively high speed
Tip of the tongue phenomenon	When a subject is able to recall parts of an item, or related information, but is frustratingly unable to recall the whole item. This is thought to be an instance of "blocking" where multiple similar memories are being recalled and interfere with each other
Trait ascription bias	The tendency for people to view themselves as relatively variable in terms of personality, behavior, and mood while viewing others as much more predictable
Travis Syndrome	Overestimating the significance of the present. It is related to the enlightenment Idea of Progress and chronological snobbery with possibly an appeal to novelty logical fallacy being part of the bias
Triviality / Parkinson's Law of	The tendency to give disproportionate weight to trivial issues. Also known as bikeshedding, this bias explains why an organization may avoid specialized or complex subjects, such as the design of a nuclear reactor, and instead focus on something easy to grasp or rewarding to the average participant, such as the design of an adjacent bike shed

Ultimate attribution error	Similar to the fundamental attribution error, in this error a person is likely to make an internal attribution to an entire group instead of the individuals within the group
Unit bias	The tendency to want to finish a given unit of a task or an item. Strong effects on the consumption of food in particular
Verbatim effect	That the "gist" of what someone has said is better remembered than the verbatim wording. This is because memories are representations, not exact copies
von Restorff effect	That an item that sticks out is more likely to be remembered than other items
Weber–Fechner law	Difficulty in comparing small differences in large quantities
Well-travelled road effect	Underestimation of the duration taken to traverse oft-traveled routes and overestimation of the duration taken to traverse less familiar routes
Women are wonderful effect	A tendency to associate more positive attributes with women than with men
Worse-than-average effect	A tendency to believe ourselves to be worse than others at tasks which are difficult
Zeigarnik effect	That uncompleted or interrupted tasks are remembered better than completed ones
Zero-risk bias	Preference for reducing a small risk to zero over a greater reduction in a larger risk
Zero-sum bias	A bias whereby a situation is incorrectly perceived to be like a zero-sum game (i.e., one person gains at the expense of another)

V Appendix 2 – Feedback Form Template

Feedback for the cognitive bias catalogue

Form description

Name

Short answer text

General comments on how you feel we can improve the content.

Long answer text

On a rating of 1 to 5(1 being lowest and 5 being highest), how would you rate the OODA class and subclass assignment for each bias?

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ Other...

On a rating of 1 to 5(1 being lowest and 5 being highest), how would you rate the examples given to understand each bias?

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ Other...

On a rating of 1 to 5(1 being lowest and 5 being highest), how would you rate the debiasing techniques for each bias?

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ Other...

VI Abbreviations

BRM – Bounded Rationality Model

DMM – Decision-making Models

REM – Rational Economic Model

RPDM – Recognition Primed Decision Model

SADD – Software Architecture Design Decisions

SDLC – Software Development Life Cycle