An Exploratory Study of Naturalistic Decision Making in Complex Software Architecture Environments

Ken Power[1][0000-0002-3782-9539] and Rebecca Wirfs-Brock[2]

[1] Independent, Galway, Ireland
ken.power@gmail.com
[2] Wirfs-Brock Associates, Sherwood, OR, USA
rebecca@wirfs-brock.com

Abstract. Architects always make decisions in some context. That context shifts and changes dynamically. Different decision-making strategies are appropriate in different contexts. Architecture decisions are at times made under conditions of time pressure, high stakes, uncertainty, and with too little information. At other times, decision-makers have sufficient time to reflect on the decision and consider alternatives. Understanding context is critical to choosing appropriate approaches to architecture decision making. Naturalistic Decision Making (NDM) explains how people make decisions under real-world conditions. This paper investigates NDM in software architecture and studies architecture decisions in their environment and decision-making context. The research approach includes a case study of large technology organizations consisting of a survey, multiple focus groups, and participant observation. Previous studies that touch on NDM in software architecture have mainly focused on decision-making processes or tools or developing decision models. This paper provides three contributions. First, we build on previous studies by other researchers to produce an in-depth exploration of NDM in the context of software architecture. We focus on Recognition-Primed Decision (RPD) making as an implementation of NDM. Second, we present an examination of the decisions made by experienced architects under conditions that can be considered *naturalistic*. Third, we provide examples and recommendations that help software architects determine when an NDM approach is appropriate for their context.

Keywords: Naturalistic Decision Making, Recognition Primed Decision Making, Software Architecture, Complexity, Decision Context, Large-Scale.

1 Introduction

Architecture decision-making is an inherently complex task because decisions often must satisfy multiple constraints and address multiple stakeholder concerns [1, 2]. Software architects make decisions related to architecture style of the system as well as technological and economical decisions [2]. Several formal, analytic architecture decision-making approaches have been published [3, 4] yet software engineering researchers find few used in practice. One explanation for this may be that complex

Power, K. and Wirfs-Brock, R. (2019) 'An Exploratory Study of Naturalistic Decision Making in Complex Software Architecture Environments', in 13th European Conference on Software Architecture (ECSA) 9-13 September 2019, Paris, France.

real-world decisions are not always about making tradeoffs, but instead about finding a decision that satisfices the current situation and allows for action [5]. Naturalistic Decision Making (NDM) originated with the goal of studying how people actually make decisions in a variety of real-world settings, as opposed to in classroom or laboratory settings [6]. These settings include conditions of time pressure, high stakes, experienced decision makers, inadequate information, ill-defined goals, poorly defined procedures, dynamic conditions, and team coordination [7]. There are times where architects need to make decisions under such circumstances. This paper seeks to further contribute to understanding how software architects make decisions under these conditions. In particular we study architecture decision-making in large, complex, software-intensive systems. Such systems are characterized by many components and sub systems developed by geographically-distributed teams, with responsibility for the architecture shared among multiple architects. Interactions among people and systems with emergent properties often result in non-linear, non-deterministic outcomes. This paper presents findings from an exploratory case study of architects making decisions in this context. Section 2 reviews key literature including a comprehensive review of the NDM literature, and studies in software architecture that mention or explore NDM. Section 3 presents the research questions and describes the approach used to answer the questions. Section 4 presents findings from this study. Section 5 is a discussion of the findings, reflecting on the research questions. Section 6 presents conclusions from this study, including a set of recommendations based on the findings, and notes future research that builds on this study.

2 Literature Review

2.1 Naturalistic Decision Making

NDM researchers specifically focus on real-world settings [8]. NDM is a "*pragmatic, realistic approach to understanding decision making*" [9]. NDM researchers have studied many settings, including firefighters, emergency responders, military personnel, police, surgeons, and design engineers [7, 10-12]. Settings under which NDM applies include the following [7]:

- **Time pressure**. NDM is concerned with how decision-makers operate when time is a constraint. Time pressure does not always mean an instantaneous response is required; NDM is cognizant of the context of the decision maker.
- **High stakes**. If a surgeon or firefighter makes a poor decision, lives can be lost. If a software architect makes a poor decision, millions of dollars can be lost. The reputation of the company and the product can be at stake.
- **Experienced decision makers**. NDM assumes experience in the domain as a prerequisite for making high-stakes decisions [7].
- **Inadequate information.** This includes uncertainty about the data, ambiguous data, and missing data. NDM researchers are interested in how decision makers

make progress in the face of the uncertainty caused by too little information, or even poor or wrong information [7].

- **Ill-defined goals**. The goal is often poorly defined or poorly structured. There is a lack of clear direction on what to do, and how to do it. The goal might change, or there could be multiple competing goals [13].
- **Poorly defined procedures**. NDM is concerned with poorly defined procedures. In contrast to conventional lab-based studies on decision-making, NDM acknowledges that decision-makers often need to invent novel procedures, or modify existing ones, in order to meet a goal [7].
- **Cue learning**. This refers to the ability of decision-makers to recognize patterns and make distinctions as an aid to decision making [7]. Building on research by Simon [14], Kahneman and Klein [15] equate this ability with intuition, noting that intuition is "*nothing more and nothing less than recognition.*"
- **Dynamic and continually changing conditions**. Decision makers need to deal with situations where the conditions around them are changing continually.
- **Group coordination**. The need for coordination among multiple people is a factor in most domains in which NDM has been studied [7].

There are many different models of NDM [16]. All these models have a purpose, and no one model encompasses everything. One of the better-known NDM models is the Recognition Primed Decision (RPD) model [17]. The RPD model focuses on assessing the situation, versus judging one option superior to others. RPD describes how people make decisions using their expertise. Experienced decision makers identify a reasonably good option as the first one they consider (cue learning), rather than generate many options for consideration. Expert decision makers conduct mental simulations of courses of action to determine if it will work, rather than contrasting strengths and weaknesses of multiple options. Where multiple options are considered, they are considered through serial satisficing rather than concurrent deliberation. An advantage of an RPD strategy is the decision maker is always ready to act, rather than waiting for a completed analysis that identifies a winner among multiple options.

The conditions under which NDM applies are, of course, not the only conditions under which architects make decisions. In analytic decision-making models the focus is on identifying situations such strategies are effective or where they fail due to cognitive limitations [18]. In contrast RPD models of decision-making focus on the conditions where people can effectively make decisions without exhaustively considering alternatives [5]. Klein identified three strategies for recognition-primed decision making: when both the details of a situation and an appropriate action are recognized, essentially an if-then-action; when an unknown situation is encountered but there are only a limited set of reasonable actions, gather and fill in enough missing information before taking an appropriate action; and when there is a known situation but the appropriate action to take is unclear, run through a mental simulation of potential actions to find the first acceptable action.

Early decision-making research focused decision-making models based on a rational consideration of alternatives. Given a known, limited set of alternatives, a decision-maker should be able reason about the alternatives. However, Simon [19] proposed that complex situations, limited time and our limited mental computational capacities constrain our decision-making and that consequently our decision-making is "bounded". Instead of collecting and processing all possible information, we necessarily construct a simplified model of the relevant factors contributing to the decision, in order to analyze the consequences of each alternative to select the "best" one. Consequently decision-making is bounded by both the structure of the information in the environment and limits of our mental capabilities [19].

Klein [6] summarizes how core beliefs in decision-making have changed. NDM asserts that experienced decision makers draw on patterns to deal with time pressure, and do not compare options. Expertise primarily depends on tacit knowledge. Projects don't always start with a clear description, particularly if dealing with "wicked problems" [20]. Experienced people in a given situation use their mental models to define what counts as data, rather than systematically building up from data to information to knowledge to understanding. Insights arise by detecting contradictions, anomalies, and connections. Uncertainty is not reduced by gathering more information but can stem from poor framing of data.

NDM research focuses on understanding the conditions under which experts make decisions and how they recognize environmental cues to guide their judgment. Skilled expertise is acquired through practice and developing skilled intuitions in high-validity environments which provide opportunities to learn [15]. Environments have high validity when there are stable relationships between cues and subsequent events, or between cues and the outcomes of actions. High validity does not correlate to certainty; some highly valid environments are also highly uncertain. Kahneman and Klein [12] observe that true experts "*know when they do not know*," but "*non-experts certainly do not know when they do not know*." The subjective confidence of a decision-maker in a decision is an unreliable indicator of a decision's validity.

Kahneman characterizes two modes of thinking: System 1, which operates automatically and quickly; and System 2, which is slower, effortful, and deliberate [21]. Both systems operate in tandem: System 1 originates impressions and feelings that are the source of beliefs and more deliberate choices made by System 2. Understanding distinctions between these systems helps inform how NDM relates to other decision making approaches [15]. For example, in Recognition-Primed Decision Making (RPDM), System 1 thinking can bring promising solutions quickly to mind, which then are simulated and more deliberately evaluated by System 2. As System 2 monitors environmental cues, System 1 intuitions may be challenged and result in more deliberate reasoning. Schraagen [22] describes the concept of 'inner' and 'outer' environments. The inner environment is about strategies and representations. Klein's Recognition-Primed Decision model is a combination of intuition and analysis [22, 23]. Recognition-based strategies enable decision makers to make decisions continuously [22].

2.2 NDM and Software Architecture

Decision-making in the field of software architecture has been the subject of study for several decades [24]. Researchers have found that most software architectural decisions are made by groups, not individuals, and that while the major factors into a decision are requirements and other constraints, architects report that personal experience and personal preference also contribute to decisions [25]. Tofan, Galster and Avgeriou [26] list 22 factors that contribute to the difficulty of architecture decisions. One of those factors is insufficient information to reduce uncertainty when making the decision. Decisions made by software architects often require consensus building and gaining trust and decisions are often made under conditions where there is insufficient information, extreme time pressures, and high stakes [27].

There has been some exploration of NDM in software architecture. This paper builds on that earlier work and contributes to a foundation on which future NDM-related research can be based. Zannier, Chiasson and Maurer [5] examine the question of how software designers make decisions. They conclude that the structure of the design problem "*as defined by the decision maker*" determines the aspects of rational and naturalistic decision-making used. Citing that paper [5], Vliet and Tang [28] study the process of making decisions in software architecture and conclude that "*… the structure of the design problem determines which aspects of rational and naturalistic decision making are used. The more structured the design decision, the less a designer considers different options.*" Context is key here, and we need to consider not just problem structure, but the context under which the designer is making the decision. Simon [29] defines a set of characteristics that determine what it means for a problem to be well structured. However, Simon [29] also warns that "*definiteness of problem structure is largely an illusion that arises when we systematically confound the idealized problem that is presented to an idealized ... problem solver with the actual problem that is to be attacked by a problem solver with limited (even if large) computational capacities.*" Here, Simon [29] (also cited by [5]) warns that definiteness around problem structure is largely an illusion, so care should be taken to not put too much effort into attempting to structure a problem definition in conditions where no such definition is possible.

Falessi et al. mention NDM in the context of comparing software architecture decision-making techniques [24]. They categorize NDM as fitting under one of three types of decision-making, where decision-makers "*keep the first available alternative.*" This is not a complete characterization of what occurs. Decision makers do not simply keep the first available option, but rather use pattern matching [30]. Falessi et al. do not mention the expert-informed pattern matching that happens. They do acknowledge the role of intuition, but not explicitly as experience-informed intuition and further characterize NDM as a decision-making technique "*where decisions are studied as the product of intuition, mental simulation, metaphor, and storytelling.*" Klein refers to these four elements as the "*sources of power*" needed in naturalistic settings [7].

Manjunath, Bhat, Shumaiev, Biesdorf and Matthes [31] mention NDM in a short paper about decision-making and cognitive biases in software architecture. They state "*evidence has been provided to show that architects either follow rationalistic or naturalistic decision-making process.*" Their reference for this statement, and their only reference for NDM, is the work by Vliet and Tang [28] in the section "*Modeling the decision-making process.*" In contrast to other NDM studies that focus on expert decision-making in context [8, 22], Manjunath et al. say, "*RPDM is derived from the naturalistic decision-making framework that relies on mental mind maps. It is generally used by inexperienced architects or in scenarios where ADDs are to be made under time pressure and other constraints which affect the decision-making quality.*" There are two potential issues with this claim. First, the primary research on NDM refers to "*mental simulation*" but does not refer to "*mental mind maps.*" Second, to say that NDM is generally used by inexperienced decision makers is not accurate. NDM emphasizes the requisite expertise of the decision maker [8, 22]. Klein further notes that differences in expertise influence decision strategy [13].

Most of these prior studies of NDM in software architecture reference one of Klein's popular books [7] or [10]. While these two books are useful, this paper cites a wider range of the NDM research literature, contributing to a deeper understanding of how architects decide and the conditions under which they make expert decisions. Other studies of NDM and architecture mentioned above examine the decision process, problem structure, or decision tools. This paper builds on these studies by focusing on the *context* of the architecture problem and the architect as decision maker in a dynamic and complex environment.

3 Research Approach

3.1 Research Setting and Context

This paper studies practicing software architects in their context. This study uses a case study of a large, global technology organization. Initially the researchers conducted an online survey of experienced architects. Of these, 70% had 6 or more years of experience as architects and were located in different sites across a global business group. The goal of the survey was to understand how architects perceived their role and interactions with other architects, engineers, product owners and product management. Following on from the survey, we conducted three focus groups to collect more data about architecture decision-making. Both the survey and focus groups targeted people with expertise in their domain, a defining characteristic of NDM settings, as discussed in section 2. The first part of this study contains details of the study design [27]. Initial observations about architecture decision-making led to a closer look at the survey and focus group data with the goal of gaining a deeper understanding of conditions and contexts under which software architects make decisions.

3.2 Research Questions

This paper is concerned with how Naturalistic Decision Making (NDM), and RPD in particular, applies to decision-making in software architecture, specifically in large and complex environments. In this context, "*large*" relates to large architectures, code bases with tens-to-hundreds of millions of lines of code, large organizations, geographically distributed teams, and products and systems developed by hundreds or thousands of engineers. "*Complex*" in this context refers to the idea that organizations are complex adaptive systems, where behavior of systems is often non-linear and non-deterministic, and the product of the interactions in the system is greater than the sum of the parts [32]. This paper aims to contribute to the body of knowledge on architecture decision-making by answering the following questions:

- **RQ1:** How does NDM apply to Software Architecture decision-making?
- **RQ2**: What are the conditions under which decisions are suited to an NDM approach in software architecture?
- **RQ3**: What are the conditions under which decisions are not suited to an NDM approach in software architecture?

3.3 Research Method

This is a qualitative study. This study uses a case study to "*understand complex social phenomena*" related to how architects make decisions. Case studies are well suited to research in software development because they study contemporary phenomena in their natural setting [33]. This study is concerned with how and why architects make the decisions they do, the context in which they make those decisions. Case studies can "*uncover subtle distinctions and provide a richness of understanding and multiple perspectives*" [34]. This research includes perspectives from multiple stakeholders, not just architects. Yin [35] notes that case studies are suitable when "*the boundaries between phenomenon and context may not be clearly evident.*"

3.4 Data Collection and Analysis

Data was collected through an online survey of 62 architects from a business group consisting of approximately 5,000 people worldwide. The survey used an online survey tool to collect responses. The researchers then followed up with three focus groups specifically about architecture decision making with 10, 11, and 12 participants, respectively, from different product lines within the business group. Participants in the focus groups were architects, program managers, engineers, and engineering managers located in Israel, the USA, and India. The focus groups were recorded, and the recordings were transcribed. The authors analyzed the survey data and focus group data independently and reviewed the analyses together through multiple iterations. Additional data was collected through participant observation and follow-up semi-structured interviews. The researchers used NVivo analyze the data.

3.5 Threats to Validity

This section discusses potential threats to the validity of this research study.

- **External Validity**. The researchers do not claim that these findings are universally applicable. They are representative of architects in specific, large global technology organizations. They serve as illustrative examples that others may learn from.
- **Construct Validity**. To mitigate this threat, data were collected from multiple sources. The researchers used triangulation between the survey data, focus groups, and participant observation, thereby converging evidence from multiple distinct data sources. The researchers compared results across multiple groups, where the data was collected at different points in time and in different geographic locations.
- **Reliability**. Relating to the repeatability of the study, the survey instrument and focus group questions were designed over several iterations and involved other subject matter experts and architects to review these and provide feedback. Using respondent validation [36] the researchers reviewed the data with a group of architects to help ensure validity of the data and the findings
- **Internal validity**. This study does not attempt to establish any causal relationships, so no internal validity threats are described [33].
- **Bias**. People tend to report decision-making experiences where there was a negative sentiment. This could impact the examples that participants chose to share. The researchers encouraged participants to consider both positive and negative experiences and outcomes.

4 Findings

4.1 NDM Conditions Under Which Architects Make Decisions

Architects in the survey report being satisfied with their decisions when they are able to share common goals, collaborate with others, and are involved early and then able follow-through their architecture decision to its implementation [30]. Feedback is important to learning. As one architect notes, "*To me it is very rewarding (for everybody) to work and agree on architecture/design decisions in order to achieve a common goal. The mutual trust and respect is very important as well.*"

When asked about challenges they faced in their role, architects expressed sentiments that exhibit several characteristics commonly found in NDM contexts. **Table 1** contains some examples of architects' experiences and how they relate to NDM characteristics. Even with extensive experience, architects don't always feel confident about their expertise. As one architect notes, "*It would be great to focus on one area for certain time to build expertise.*" Here, they are referring to a particular type of expertise, i.e., expertise in the product, system, or subsystem. Dynamic, shifting responsibilities, and changing business demands added to their stress and lack of confidence in their decision-making abilities.

4.2 How Attributes of NDM Decision Making Influence Decision Making

Focus group participants were asked to share their experiences of architecture decisions that they were involved with. A significant number of the examples from the focus groups show evidence of conditions typical of NDM settings as characterized in section 2. Findings are presented here in the same order as the NDM settings in section 2.2.

Table 1. Examples of selected NDM characteristics from the study findings; TC=Team Coordination, II = Inadequate Information, TP= Time Pressures, PDP= Poorly Defined Procedures, DCC= Dynamic Changing Conditions

Architect's Experience	**NDM Characteristic**				
	TC	II	TP	PDP	DCC
"finding time for direct collaboration in calendars."	X		X		
"Getting enough time from the knowledgeable architects is difficult - especially when their agendas are not completely aligned with mine.	X	X	X		
"Time – we're all busy!"			X		
"we spend a lot of time in discussions and speculations of how a feature was designed and implemented, instead of referring to a system spec"		X			
"The transition to feature teams has dissipated in-depth knowledge of our software"		X			X
"Not all the information is shared with architects which could affect some architecture decisions in the initial phase of the project."		X			
"Not being aware of system-wide decisions (guidelines, policies) until long after they are made"		X			
"It would make my job easier if other architects would be concerned with making sure that others know what they are working on, what decisions they have made that affect my work."		X	X		
"Without an agreed process, there is always the tension between the fast and dirty guys and the more structured guys who keep records of requirements and design."				X	
"The developers are encouraged by their managers to provide independent solutions without seeking for an agreed design, and sometimes even against an agreed design."			X	X	
"Feedback on architectural decisions takes years, if ever, to arrive. This makes learning from experience difficult if not impossible"		X			
"Our organization has been in firefighting mode for a long time, and that inhibits the ability to take a step back and look at the bigger picture."					X
"People are so insecure about their jobs … that they are protecting information, not sharing, and are not open to	X	X			X

suggestions."					

Time Pressure. The findings show examples of decisions that were made under time pressure. One architect told of a decision made to implement a simple coding change, even though it was known to be inadequate at the time and other alternatives could have been explored. The reason for accepting the solution was, *"because it was urgent. Right now."* Another architect told of being directed to change their design to "*just make it fit*" time allotted. Although architects acknowledged that decisions need to be made for short-term expediency, e.g. to address an immediate customer need, they aren't always happy about it. One architect described frustrations felt about a decision where, "*The right people were in the room, but there were arguments that were raised for the first time during this meeting. And we came with a proposal, and for some reason, during a very short discussion there were raised new arguments that couldn't be assessed properly. And I think that there was a need to stop the discussion, go and analyze the feedback, but was under a lot of ... I'm not sure if it was the real pressure, but we wanted to finish up and to get a decision, and I'm not sure that the right decision has been taken, just because of lack of time.*" Was it actually lack of time, or perceived lack of time? Is there a difference in how action is taken? If the decision-maker feels time pressure, then it is real for them. Options are narrowed when under time pressure. There is also evidence of decisions that were not made under time pressure. Architects shared examples of decisions that were technology focused and strategic or long-term in nature, e.g., API evolution, or creating guidelines for the use of microservice frameworks. These decisions were made more deliberately, involving experimentation and analysis.

High Stakes. The financial stakes are only one perspective of architecture decisions. Architects in this study make decisions that impact products and systems with multi-million- and multi-billion-dollar revenue streams. However, high stakes are not just because of financial concerns. Architects make decisions that impact customer relationships, company reputation, future evolvability of the architecture, and market competitiveness. Trust among peers and colleagues is a further theme that emerged; the stakes are also high if that trust can be damaged. A discussion on how to establish that trust and mutual respect is beyond the scope of this paper.

Experienced decision makers. The architects in our study were experienced and generally confident about their decisions. However, occasionally they encountered situations where they felt they lacked expertise. For example, one architect recounted several situations where teams came to him for decisions even though he was not an expert in their particular product area. He expressed feelings of self-doubt *("I don't know all of these things. Winging it most of the time. I don't really understand a lot of this stuff.").* As the discussion progressed it was clear that the architect was an experienced architect, and familiar with the technology domain. What he felt he lacked was specific experience with the technical components the teams needed help with, which were outside his immediate scope of responsibility. However, his general expertise as an architect and his expertise with the domain resulted in “good enough” decisions that got the team over their immediate hurdle. This also bought them time to fill the gap in organization knowledge. In another situation an architect explicitly sought expertise, in order to make better-informed decisions: *“For example, my team was doing a feature …. They made a lot of changes we’re not still really comfortable about, and then …we went to approach the guy who had left our team. So, he came and he was the one who reviewed.”* Developing expertise takes time. An experience was shared of a team that deliberately acquires necessary expertise to competently make decisions in new domains: *“Basically, when they become incompetent, they just close the doors. And they say, ‘We will not entertain any request on this component for the next six months. Nothing. Don’t come and talk to us if you want us to do a good job…’. …[And on] the code, they write test cases, they reverse engineer, the whole thing. Then they come back six months later, and it really is like you just changed into a butterfly from a caterpillar. … At which point, they’re really good.”*

Inadequate information. Finding information can be difficult as one architect observed: “so much documentation is missing that … it becomes very complex to go through the code and do the reverse engineering of what was thought.” Yet not every architect expects important details to be recorded: *“Usually, the decision of what was decided will be captured in the document. The decision of why it was decided that way should be captured in somebody’s head.”* There are counter examples in the findings where design rationale was documented: *“They do use Confluence for managing everything about the…decision to be taken and conclusions and conversations and thoughts around the decision and everything we’ve documented, easily to be accessed again… I'm using these sites and these pages. I always find what I’m looking for.”*

Ill-defined goals. Designing the high-level architecture for a feature can be complicated as an architect notes: “*Just the countless numbers of architects that are involved and the lack of clear product ownership because we moved away from component ownership to this feature ownership. Which the lines become blurred because you can own a feature and you’re shifting a feature into the solution but then it may impact a number of other things supporting related features and stuff. It’s hard to understand where’s the start and end of the product that you’re supposed to be driving.”*

Poorly-defined procedures. One architect expressed uncertainty about who should be involved in decision-making: *“I think that job description or responsibilities are not well defined. Therefore, I’m talking about myself, you can always ask yourself whether you are the right person to take [a] decision or do you need to consult with someone else, or are you stepping on someone’s toes or not.”* The shift to agile development has made the process of architecting system infrastructure less obvious; as an architect noted: "*Agile hasn't given an adequate answer to scaffolding or to infrastructure... So as long as we're talking a feature which has some kind of huge impact it's OK. If it's very narrow, end to end, it doesn't impact on the system then it's fine.*"

Cue Learning. An architect brought up the issue that sometimes short-term decisions may not be revisited, even when evidence may indicate that this would be judicious: *"The problem is that when what I think is decision making in many, many cases the first decision is accepted as the final one and the project leader [is] not ready to change direction and adjust the decision to problems found."* The discussion continued around what to do with new evidence, as the architecture can't always be in flux. As one architect notes, *"People, in order to develop a solution, in order to develop interfaces, in order to ... They need some stability. Even if it's not the ideal solution, we need a consistent solution."*

Dynamic and continually changing conditions. Under pressure to decide, one architect stated they had difficulty finding consensus for the bigger decisions that needed to be made, *"just because things are moving too fast for me, and the organization is too in flux."* Consequently, they made lots of shorter-term decisions to compensate. This person was an experienced architect and recognized the need for considering long-term impacts. Under conditions of uncertainty and time pressure they adopted a strategy that would be good enough in the short term, and keep the team moving towards their longer-term needs. Another architect shared that they adjust their initial decisions based on direct feedback and changing conditions. There are also examples where organization politics can influence decision-making, adding to the volatility and uncertainty of the context. For example, an architect shared that a directive was given and not challenged: *"my feeling was even though that was a directive or decision, not enough attention was given to nuance and to actual issues that will arise from the deployment."* Another architect stated that "*there are cases when up to discussion, the people who disagree with mainstream were removed from the discussion."*

Group coordination. While not all decisions are made by consensus, it often takes time to gain consensus. One architect notes *"because we're focused on consensus over multiple engineering teams and architecture teams all over the place, the process has just gotten more complicated.*" Another architect remarks *"To me [it] is very rewarding (for everybody) to work and agree on architecture/design decisions in order to achieve a common goal."*

5 Reflections on the Research Questions

RQ1: How does NDM apply to Software Architecture decision-making?

The study found that experienced software architects make many decisions under dynamically changing business conditions, with time pressure, and having inadequate information. NDM, and RPD in particular, seems suited to decisions that must be made quickly and when fast feedback on the decision allows for course corrections. The conditions under which NDM is appropriate, however, can be short-lived, e.g., the time pressure is temporary. Goals can become clear, or the need for clarity passes. Information becomes available, or the need for that information passes.

RQ2: What are the conditions under which decisions are suited to an NDM approach in software architecture?

Decisions that are made collaboratively, where there is mutual respect and trust among decision-makers, and there's enough expertise seem to be well suited to NDM. Kahneman and Klein also observe that true experts know when they don't know and that ability to recognize a situation as novel is one of the characteristics of experts [15]. The case study found examples where architects who didn't know enough to take a decision with confidence either found a way to limit the scope of a decision to what they felt expert in, or found and utilized others' expertise to improve the decision. Given the complexities of the systems they are designing, architects feel more confident in their decision-making when they can learn from engineers and receive feedback on the implementation of their decisions.

RQ3: What are the conditions under which decisions are not suited to an NDM approach in software architecture?

Decisions that require investigation into new technologies or are outside the area of expertise of an architect are not suited to NDM approaches. Other examples where more analytic approaches are appropriate include choosing a new persistence technology or migrating to a microservice architecture. These are conditions where poor information is not tolerable. Environments where there are panels for reviewing architectures generally won't use NDM approaches. In these settings, decisions are made through argumentation, persuasion, and influence – tactics for which there is rarely time in NDM settings. Moreover, even though certain situations may appear conducive to NDM approaches, architects themselves, may question their own expertise, and thus may seek out advice or take a more analytical approach to making an architecture decision.

6 Conclusions

NDM is not a design decision process, but a way of understanding the context in which decisions are made that, in the context of this study, impact architecture. NDM is therefore process-agnostic. This exploratory study concurs with the findings of Klein [17]; namely that recognition primed decisions are more likely when the decision maker is experienced in the domain, time pressure is great, and conditions are less stable. It can be helpful to consider decisions as related to three domains of technology, solution, and product [27]. NDM is more likely to apply to select decisions where new technologies are being introduced. NDM also applies in situations where business and solution contexts are poorly understood, or are being invented, and this has an immediate impact on architecture. Market and competitive pressures can force situations that benefit from NDM. The team needs to decide something quickly and move on. We found evidence that architects learn under conditions of uncertainty when they get feedback. This feedback adds to their expertise and contributes to their learning of important cues. This improves their capability for dealing with future scenarios where recognition-primed decision making is important. These findings are in contrast to other researchers who claim that "*RPDM is generally used by inexperienced architects*" [31]. Working in a complex, distributed environment poses great challenges for naturalistic forms of decision-making. It can be difficult to get meaningful and timely feedback. Decisions that involve a larger group take more time and consensus building. One strategy reported to speed up decision-making was taking decisions that were more limited in scope instead of building consensus. We also found examples where decision-makers, when they felt they lacked expertise, found other experts to help in making decisions or took the time to develop necessary expertise before taking any actions. We also found an example of an architect who was called on to make decisions because he was perceived as being good at making decisions, even though he lacked specific expertise. Our findings concurred with the NDM literature that there often is not enough time to build trust or gain widespread consensus. We observed that authority is granted to architecture decision-makers based on expertise and role. There is often an implicit and immediate and unspoken agreement on granting this trust and authority in a triage situation that requires a rapid architecture decision.

6.1 Recommendations

Based on the findings in this exploratory study into NDM and architecture, the researchers propose the following preliminary recommendations for architects and those responsible for creating the conditions under which architects do their work:

- Experts may not retain their tacit knowledge-informed expert status under dynamically changing conditions. They may quickly and temporarily find themselves operating in an environment where their particular expertise does not apply. Architecture expertise needs to be refreshed in software architecture. Architects are not just doing the same thing over and over again. The context is shifting. Consequently, a lot of learning happens on the job and timely feedback is essential to learning.

- Consider carefully the consequences of using NDM approaches when the necessary expertise is lacking. Expertise is a critical factor to successful decision-making. Growing expertise requires feedback on the consequences of decisions and collaboration with others to share knowledge.
- Most architectural decisions are group decisions. NDM is more challenging in the context of large groups that are distributed. More formality may be required to reach agreement and document decisions in such settings.
- NDM decision-making may not be appropriate for locally optimized architectural decisions. Sometimes seemingly localized decisions have broad system impacts. In these situations, analytic approaches to decision-making may be more appropriate.

6.2 Future Research

This paper describes the first steps in a series of studies that the researchers are working on towards understanding how the software architecture profession can benefit from understanding software architecture through the lens of naturalistic decision making. This has applications for architects, architecting, and architecture. A better understanding of Recognition Primed decisions (RPD) and other NDM models will help architects apply appropriate decision-making strategies in the right context.

While localized decisions may appear expedient, sometimes they can have a broader impact than anticipated. Understanding what conditions under which narrower decision-making contexts are appropriate as well as the potential impacts of a series of micro decisions is a topic of future research.

In addition, the social and political influence on decisions emerged as a point of interest from these findings and is an area worthy of exploring in the context of NDM. The NDM literature says little about the social and political context, e.g., they don't talk about politics of hospitals or fire stations. They focus on expertise. However, in real-world software organizations, political factors are also an influence on decisions.

Klein, Ross, Moon, Klein, Hoffman and Hollnagel [36] report that as people gain experience, they spend more time examining the situation and less on contrasting the options, whereas novices spend more time contrasting options and less on comprehending the situation. We didn't find evidence to support or disprove this finding, as the architects in our study weren't novices; they were experts encountering novel conditions where they needed to make decisions. Further research is needed into how experienced architects approach decision-making under novel conditions.

References

1. Dasanayake, S., Markkula, J., Aaramaa, S., Oivo, M.: Software Architecture Decision-Making Practices and Challenges. 24th Australasian SW Eng Conf, pp. 88-97. (2015)

2. Rekha V., S., Muccini, H.: Suitability of Software Architecture Decision Making Methods for Group Decisions. Software Architecture, pp. 17-32. Springer (2014)
3. Zalewski, A., Kijas, S.: Beyond ATAM: Early architecture evaluation method for large-scale distributed systems. Journal of Systems and Software 86, 683-697 (2013)
4. Cervantes, H., Kazman, R.: Designing Software Architectures: A practical approach. Addison-Wesley, Boston, MA (2016)
5. Zannier, C., Chiasson, M., Maurer, F.: A model of design decision making based on empirical results of interviews with software designers. Info. and Software Tech. 49, 637-653 (2007)
6. Klein, G.: Reflections on applications of naturalistic decision making. Journal of Occupational and Organizational Psychology 88, 382-386 (2015)
7. Klein, G.A.: Sources of power. MIT Press, Cambridge, MA (2017)
8. Klein, G.: A naturalistic decision making perspective on studying intuitive decision making. Journal of Applied Research in Memory and Cognition 4, 164-168 (2015)
9. Gore, J., Banks, A., Millward, L., Kyriakidou, O.: Naturalistic Decision Making and Organizations: Reviewing Pragmatic Science. Organization Studies 27, 925-942 (2006)
10. Klein, G.A.: Streetlights and shadows : searching for the keys to adaptive decision making. MIT Press, Cambridge, MA (2009)
11. Klein, G.: Naturalistic decision making. Human factors 50, 456-460 (2008)
12. Klein, G.A., Calderwood, R.: Decision models: Some lessons from the field. IEEE Transactions on Systems, Man, and Cybernetics 21, 1018-1026 (1991)
13. Klein, G.: Naturalistic Decision Making. Human Systems IAC XI, 16-19 (1991)
14. Simon, H.A.: What is an explanation of behavior? Psych. Sci. 3, 150-161 (1992)
15. Kahneman, D., Klein, G.: Conditions for intuitive expertise: a failure to disagree. American psychologist 64, 515 (2009)
16. Flin, R., Salas, E., Strub, M., Martin, L.: Decision Making Under Stress: emerging themes and applications. Routledge, New York, NY (2016)
17. Klein, G.: A Recognition-Primed Decision (RPD) Model of Rapid Decision Making. In: Klein, G.A., Orasanu, J., Calderwood, R., Zsambok, C.E. (eds.) Decision Making in Action: Models and Methods, pp. 138-147. Ablex Publishing, Norwood, NJ (1993)
18. Kahneman, D., Tversky, A.: Choices, Values, and Frames. Am Psy 39, 341-350 (1984)
19. Simon, H.A.: Invariants of Human Behavior. Annu Rev Psychol 41, 1-19 (1990)
20. Sherman, R.O.: Wicked Problems. Nurse Leader 14, 380-381 (2016)
21. Kahneman, D., Egan, P.: Thinking, fast and slow (2011)
22. Schraagen, J.M.: Naturalistic Decision Making. In: Ball, L.J., Thompson, V.A. (eds.) International Handbook of Thinking and Reasoning, pp. 487-501 (2017)
23. Klein, G., Calderwood, R., Clinton-Cirocco, A.: Rapid decision making on the fire ground: The original study plus a postscript. J of Cog Eng and Decision Making 4, 186-209 (2010)
24. Falessi, D., Cantone, G., Kazman, R., Kruchten, P.: Decision-making techniques for software architecture design. ACM Computing Surveys 43, 1-28 (2011)
25. Miesbauer, C., Weinreich, R.: Classification of Design Decisions – An Expert Survey in Practice. In: Drira, K. (ed.) Software Architecture: 7th European Conference, ECSA 2013, vol. LNCS 7957, pp. 130-145. Springer, Montpellier, France (2013)
26. Tofan, D., Galster, M., Avgeriou, P.: Difficulty of Architectural Decisions – A Survey with Professional Architects. In: Drira, K. (ed.) Software Architecture: 7th European Conference, ECSA 2013 Proceedings, vol. LNCS 7957, pp. 192-199. Springer, Montpellier, France (2013)

Power, K. and Wirfs-Brock, R. (2019) 'An Exploratory Study of Naturalistic Decision Making in Complex Software Architecture Environments', in 13th European Conference on Software Architecture (ECSA) 9-13 September 2019, Paris, France.

27. Power, K., Wirfs-Brock, R.: Understanding Architecture Decisions in Context: An industry case study of architects' decision-making context. In: Benedí, J.P., Cuesta, C.E., Garlan, D. (eds.) Software Architecture. Springer (2018)
28. Vliet, H.v., Tang, A.: Decision making in software architecture. JSS 117, 638-644 (2016)
29. Simon, H.A.: The structure of ill structured problems. AI 4, 181-201 (1973)
30. Klein, G.: Naturalistic decision making: Implications for design. KLEIN ASSOCIATES INC FAIRBORN OH (1993)
31. Manjunath, A., Bhat, M., Shumaiev, K., Biesdorf, A., Matthes, F.: Decision making and cognitive biases in designing software architectures. 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), pp. 52-55. Seattle, WA, USA (2018)
32. Bohórquez Arévalo, L.E., Espinosa, A.: Theoretical approaches to managing complexity in organizations: A comparative analysis. Estudios Gerenciales 31, 20-29 (2015)
33. Runeson, P., Höst, M., Rainer, A., Regnell, B.: Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons, Inc., Hoboken, NJ (2012)
34. Kohn, L.T.: Methods in Case Study Analysis. Technical Publication, Center for Studying Health System Change (1997)
35. Yin, R.K.: Case study research : design and methods, 5th Edition. SAGE, London (2014)
36. Klein, G., Ross, K.G., Moon, B.M., Klein, D.E., Hoffman, R.R., Hollnagel, E.: Macrocognition. IEEE intelligent systems 18, 81-85 (2003)

Power, K. and Wirfs-Brock, R. (2019) 'An Exploratory Study of Naturalistic Decision Making in Complex Software Architecture Environments', in 13th European Conference on Software Architecture (ECSA) 9-13 September 2019, Paris, France.