2017 IEEE International Conference on Software Architecture

Human Aspects in Software Architecture Decision Making

A Literature Review

Antony Tang
Swinburne University of Technology
Melbourne, Australia
atang@swin.edu.au

Maryam Razavian
Technische Universiteit Eindhoven
Eindhoven, The Netherlands
m.razavian@tue.nl

Barbara Paech, Tom-Michael Hesse
Heidelberg University
Heidelberg, Germany
{paech,hesse}@informatik.uni-heidelberg.de

***Abstract*—Despite past efforts, we have little understanding and limited research efforts on how architects make decisions in the real-world settings. It seems that software architecture researchers make implicit assumption that decision making by software architects can be a rational and prescribed process. Such an assumption is disputed in other fields such as economics and decision research. This paper studies the current state of software architecture decision making research in terms of human behaviors and practice. We carried out a literature review on software architecture decision making. We classified papers into decision making behavior and decision making practice and identified the research relationships between them. We found that decision making is a mental activity. Research into the behavioral aspects of software architecture decision making for incorporation into architectural design practices is required. We suggest three research topics on human aspects to improve software architecture practices.**

Keywords- software architecture; decision making; human behavior; methods and tools;

I. Introduction

For nearly two decades, there has been much interest in the software architecture community to explore how design rationale [1-3] and software knowledge management [4, 5] help software architecture design. The basic premise of such approaches is that knowledge and rationale give additional information and argumentation in designing. However, the decision making mechanism is not very well understood in software architecture. Researchers in the software architecture field may have overlooked factors such as biases and group dynamics that influence software decision making [6], and we want to investigate what has been done.

Software designers and developers make decisions regularly, even though they may not be aware of how they make decisions. For instance, they make decisions on what architecture style to use, how to design an API, or what methods should be included in a class. Software architecture decisions are more than the synthesis of knowledge and information into software outcomes, justified by some design rationale. The process of software architecture design involves many stakeholders and a wide range of activities that includes defining goals, defining and clarifying requirements, defining software structures at abstract and code levels. All these activities involve decision making [7]. Do software architects make good decisions when given the right information? Is one way of decision making better than another way? Is there a better way, under given circumstances, of making good decisions? We generally know that sound decision making underpins the quality of good software systems, but we do not really know how to achieve sound decision making.

Classical economics theories make assumptions about how consumers make choices from optimal beliefs and rationale [8]. Researchers later found that other forces, such as bounded rationality can influence decision making [9]. The long-held assumption of having full market knowledge and rational choices to optimize economic decisions does not hold anymore. Consumer rationality and cognitive biases need to be taken into consideration in economic theories, thereby forming the basis of behavioral economics [8]. Decision researchers suggested that it is not obvious how decision makers make decisions. Sometimes decision makers themselves cannot tell how they make decisions. So it is important to investigate how decisions are made and how to improve decision making [10]. Software architecture researchers have investigated how to aid decision making. However, some architects and researchers make the implicit assumption that software design can be a rational and explicit process. This assumption is questionable. In software engineering generally, human aspects of decision making have been recognized as important but not often considered [11].

Software architecture decision making is more than mechanically applying some prescribed methods. Humans are involved and humans make decisions in different ways. We need to understand the human aspects of decision making and to gain more insights to how software architects design. But first, we wish to do an inventory of how much we know about software architecture decision making. To do so, we study empirical research works on decision making. We ask this research question (RQ):

RQ: What research on human aspects of software architecture decision-making has been done and how does it reflect on software architecture decision making?

Rationale: Human is a first-class entity in software architecture decision making driven by some internal behavioral processes. Decision making is influenced by engineering processes and methods that are practiced. Our

978-1-5090-5729-0/17 $31.00 © 2017 IEEE
DOI 10.1109/ICSA.2017.15

research focuses on these two aspects. We select empirical research papers in this study because the study of human aspects requires empirical evidence and cannot be anecdotal [12]. This study allows us to (a) gain an overview of the subject; and (b) analyze the research directions in this area.

Our research approach is to analyze software architecture decision making research papers. In Section 2 we describe our literature review and analysis procedure. We summarize the research results in Section 3. We interpret these findings and discuss the implications for future research in Section 4.

II. Literature Review

A. Literature Identification and Selection

This research study software architecture decision making research. We take a broad view of software architecture that includes software requirements and design. We started by collecting software decision making research papers that are known to us (Step 1 in Fig 1). From these papers, we identified eight venues that are likely to contain such research works (Step 2). We considered them as the *primary sources*. These are: (a) Journal of Information and Software Technology (IST); (b) Journal of Design Studies (JDS); (c) Workshop on Sharing and Reusing Architectural Knowledge (SHARK); (d) IEEE Software; (e) Journal of Systems and Software (JSS); (f) Quality of Software Architecture (QoSA); (g) Working IEEE/IFIP Conference on Software Architecture (WICSA); (h) European Conference on Software Architecture (ECSA). Seven of the eight venues are where software architecture researchers often publish their works with the exception of JDS. We picked JDS because it has a focus on design and there were a number of studies of how software designers think in a special issue [13]. We manually traversed the past issues of these eight software venues to find relevant papers (Step 3). We looked through all issues for the 11 years from 2005 to 2015. The reason for selecting 2005 is because at that time design rationale study started to take off in the software architecture field with prominent research papers such as [2, 14]. We started this research in early 2016 and hence we finish our literature review by end of 2015. We retrieved research papers from these primary sources through manually reading the paper titles and abstracts from these eight venues (Step 3). We examined the title and the abstract and looked for key phrases like "decision", "design decision" or "decision making".

There are also ***secondary sources*** where we found relevant research papers. With the results from our search in the primary sources and the known papers (Step 4), we used a snowballing technique (Step 5) [15] to find relevant papers from citations. We also included papers that we know about before this review, some of them are well-known papers dated earlier than 2005 (Step 1). These papers are from Empirical Software Engineering, IEEE Expert, Communications of ACM, ACM Computing Survey, International Journal of Human-Computer Interaction, IEEE Transaction of Software Engineering, Agile Conference, Automated Software Engineering and book chapters (Step 5). We ended up with a preliminary set of seventy-seven (77) papers from both sources. During Step 5, we also found twelve (12) research papers that are relevant to software decision making, but the subject of the study is not software development. Based on this set of papers, we selected research papers to be included in our analysis through applying ***selection criteria*** (Step 6). First, a selected paper must study one of the two subjects: (a) factors that affect software decision making, especially human factors; (b) software decision making practice in a software development environment. Second, a selected paper must have conducted research to yield empirical results. This criterion eliminates papers that are anecdotal or survey in nature. Third, if a paper does not relate the research results to software decision making directly, the work is excluded from our review.

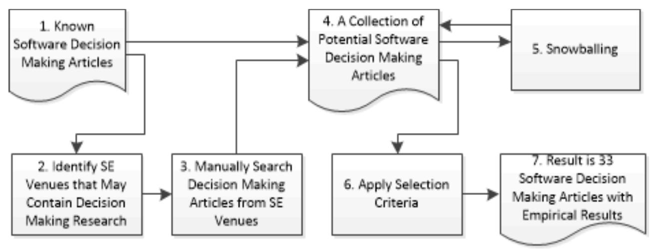

Figure 1. Paper Selection Process

Four researchers were involved in reading the papers. We arranged the reading, selection and coding of the papers such that (a) each paper was assigned randomly and read and coded by two researchers; (b) each researcher had to determine if the paper fits the inclusion and exclusion criteria; (c) each researcher read at least forty-five (45) papers; (d) a researcher cannot code or select the paper s/he wrote. We have finally selected a total of thirty-three (33) papers. Table 1 summarizes the search results. The columns indicate at which stage the papers were identified and if a paper is selected or not. ***Step x in*** signifies that a paper from a particular paper source passes the selection criteria. For instance, cell "*Step1 in / IST*" shows that the *known software decision article* (Step 1) in IST have been selected after applying the selection criterion in Step 6. ***Step x out*** shows papers that do not meet the selection criteria. S_i in each cell is the paper identifier.

We coded each paper with a summary and general assessment and details on the mentioned humans aspects and practice: the overall strategy (naturalistic or rational), cognitive aspects (in particular familiarity, expertise, bias), process aspects (decision making task, artefacts, tools, methods, constraints), decision making activities (creation, review or evolution of a decision) and sub-activities (determination, structuring, discussion; explicit/ implicit decisions), decision knowledge (such as problem, solution, context, rational, external knowledge and their relations). We also coded the empirical study method, but this is not used in this paper. When the codes of the 2 researchers differ we discussed and made adjustment.

TABLE I. RESULTS OF PAPER SELECTION

	Step1 in	Step1 out	Step3 in	Step3 out	Step5 in	Step5 out
IST	S46,S80		S3, S52	S38		
JDS	S7,S66					S4
SHARK	S81	S65				
IEEE S/W		S74,S83		S23		
JSS	S33,S35	S84,,S68	S21, S39	S59,S75		
QoSA	S69	S42				
WICSA	S30,S32, S47,S60, S67	S29,S37, S85	S19, S26, S61	S11,S13, S64,S71		
ECSA	S51,S70	S48	S31, S73	S1,S16,S22 ,S36,S58, S72,S76		
Others	S9,S27, S28,S49, S53	S5,S12, S82			S14, S15, S62, S77	S2, S10,S17, S18,S24,S4 0,S43,S44, S50,S55, S56,S63, S86,S87, S88,S89

B. *Limitations*

We did not carry out a systematic literature review by searching all databases. This means that we may omit papers from other venues (such as CHASE workshop). We are limited in our claims that all software architecture decision making literature is included. We only surveyed the mainstream software architecture research venues that are likely to publish such works. We judge that we have a fair representation of the publications on software architecture decision making, and our method is rigorous. As part of the review, we also gather research papers from other disciplines, notably psychology, cognitive science, and design studies to enhance our understanding of decision making in general. We did not carry out any comprehensive literature search in these other disciplines. We followed the citations from the software papers we found to seek out these useful papers from the other disciplines. The research works from the other disciplines provide ideas and lessons for software researchers, to study software decision research and research methodologies. These papers were referenced in this paper to give us relevant background materials.

III. SOFTWARE ARCHITECTURE DECISION MAKING RESEARCH

We found thirty-three papers that are concerned with software decision making with empirical evidence. According to our selection criteria on the subject, we first classified whether a paper is focusing on *human decision making behavior* or *decision making process, tools or methods*. Respectively, we classified eleven papers into **DMBehavior** and the other twenty-two papers on software architecture decision making techniques, methods and tools into **DMPractice**. These papers generally observe decision making activities or they test process/methods/tools to improve decision making. We further sub-classified these papers based on our coding. We show these two main classes of papers (DMBehavior and DMPractice) in Fig 2, the number of papers found in each sub-class (i.e. shown within the brackets) and the identified papers in each sub-class.

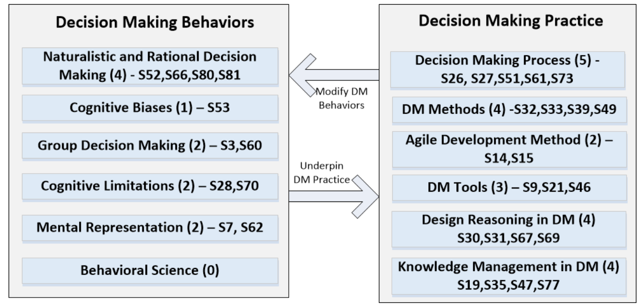

Figure 2. Decision Making Research Classification

The coding and subsequent classification were based on the main goals and findings of each paper. For instance, **S80**'s main finding was naturalistic decision making behavior, so we created a sub-classification for this type of research. Some papers have findings that can belong to more than one sub-class. We classify such a paper into one class only based on the main result. This simplification gives us a better view of the current state of research works. There are 6 sub-classes in DMBehavior and 6 sub-classes in DMPractice. The details of the findings in each class are explained in Section IIIA and IIIB, respectively.

A. *Decision Making Behaviors*

Eleven DMBehavior papers studied psychological and cognitive aspects of decision making and they deal with different human thinking aspects. We found, 5 sub-classes according to the primary subjects of these 11 papers. We found 4 papers that study *naturalistic and rational* decision making. One paper dealt with *cognitive bias*. Two papers studied *Group decision making*. Two papers studied cognitive limitation and satisficing behavior, classified as *cognitive limitation*. Finally, two papers studied mental characteristics and experience, they were classified as *mental representation* papers. There is a group where no papers were found. It is about decision making behaviors, and we call it *behavioral science* papers. Behavioral science is one of the psychology areas that are widely studied in management and organizations [10]. There is an awareness and studies of behavioral science and decision making in information system field [16]. In our review, we have found no works that investigate organization behaviors, motivations, or personality with respect to software decision making. The number of behavior science papers shown in Fig. 2 is zero. Although no such papers were found in the software architecture field, we report this category because other disciplines have shown that these are important contributing factors to decision making. It would be a major gap in this classification if we omit it.

1. Naturalistic and Rational Decision Making - In naturalistic decision making (NDM), people frequently construct explanations of decisions in the form of stories about possible outcomes. Naturalistic approaches to decision making are more contextually embedded, subjective, and stress the roles of identity and unconscious emotions in decision making [6]. Rational decision making (RDM) describes how decision makers think and act based on coherence and rationality. A decision maker optimizes decisions between choices of alternatives in well-structured settings. Kahneman uses the terms System 1 and System 2 thinking. System 1 is fast, instinctive and emotional, and evolutionary very old. System 2 is slower, more deliberative, and more logical, and evolutionary more recent [17]. In this sub-class, we include papers that reference any of the two theories, i.e. System1/System 2 or NDM/RDM.

We found 4 papers that base their arguments on either of these two systems of decision making. **S52** did a multi case study of agile teams and found that the teams used NDM. **S66** studied how software designers explore the problem and solution space and the role of reasoning in decision making. It was found that explicit reasoning helps designers to communicate better and to avoid assumptions in decision making. It was suggested that System 2 helps problem space exploration and considerations of solution alternatives. **S80** studied decision making of 25 practitioners. It was suggested that the more structured is the problem, the more RDM is used, and the less structured is the problem, the more NDM is used. **S81** conducted three case studies and found that designers do not consistently strive for optimal design solutions, which is a key characteristic of RDM.

2. Cognitive Biases - Cognitive bias is the general term, introduced by Kahneman and Tversky [18], to denote human's inability to reason rationally. They are cognitive or mental behaviors that prejudice decision quality in a significant number of decisions for a significant number of people [19]. In our study, we found one DMBehavior paper that researched into cognitive bias. In **S53**, the researchers conducted a controlled experiment to explore the relation between the design process and the framing/presentation of requirements. It was found that framing desiderata as "requirements" negatively affect creativity in design concept generation, indicating that the term requirement may curtail innovation independent of the requirements specifications themselves.

3. Group Decision Making - Software decisions are often made in a group environment. Different group decision making (GDM) tactics such as majority rule, plurality rule and Condorcet winner are discussed in [20]. Two DMBehavior papers studied group thinking in SE. **S3** focused on the understanding of how software professionals in groups invoke knowledge in their communication, reasoning and decision making for software effort estimation. Using planning poker, the researchers found that concepts used in estimation are anchored in the software engineering knowledge domain and in historical experiences of the participants. Knowledge is constructed with a basis in social interaction, drawing on specialized concepts from the knowledge domain of software systems in the participants' efforts to frame and guide the talk. **S60** investigated GDM using an online survey with practitioners and researchers in the software architecture community. They found that consensus and brain storming are used in 70% of companies. AHP, Delphi and voting methods are used by 50% of companies. They also identified group decision challenges: (a) groupthink when the group structure is highly cohesive; (b) misunderstanding of goals; (c) conflicting decisions.

4. Cognitive Limitation - Cognitive limitation refers to the limitation in the capacity of short-term memory or unreliable retrieval of relevant information from long-term memory [21]. In decision making, rationality of individuals is limited by the information they have, the cognitive limitations of their minds, and the finite amount of time they have to make a decision [22]. Due to cognitive limitations and other constraints such as time, decisions are made without thorough reasoning, and decision makers satisfice with decisions. Satisficing indicates that a decision maker makes a decision that is good enough to satisfy the goals, and the decision maker seeks a satisfactory solution rather than an optimal solution [23].

Two papers were found. **S28** reported problem solving of 8 professional programmers. Researchers observed and analyzed what breakdowns, or difficulties, the professionals encountered. The study found several breakdowns: (a) difficulty in considering all stated and inferred constraints in a solution; (b) difficulty in performing complex mental simulations with many steps or with many test cases; (c) difficulty in keeping track and returning to aspects of problems whose solution refinements has been postponed; (d) difficulty in expanding or merging partial solutions into a complete solution. **S70** investigated the extent to which students and professionals look for alternatives in design decision making. It was found that most designers make decisions when they found good enough reasons. No thorough explorations and reasoning were performed and not many options were explored.

5. Mental Representation - When a designer solves a problem, the problem is mentally structured and transformed into a representation of the current situation and goals. An understanding of how goals, problems and other relevant information are arranged and processed mentally gives us insights on how decisions are made. Many studies compare how experts and novices perform the same tasks by comparing their cognitive characteristics and mental representation. Expert mental representations were found to demonstrate superior extent, depth and level of details [24]. Experts accommodate information interconnections and gear decisions towards actions. They view problems as harder than novices in that experts reported needing more information in order to tackle problems. Experts demonstrated more depth and width in the scope of their mental representations.

We found two papers that dealt with the issue of mental representation and cognitive characteristics in software decision making. **S7** found that software designers use a creative cognitive process to explore and generate in a sequential way, starting with an extensive use of exploratory tasks such as hypothesis testing and functional inference exploration and through that come up with generative ideas like associations and analogical transfer. In decision strategy, software designers used stepwise refinement, in which a complex design problem is decomposed top-down into sub-problems. **S62** studied the cognitive characteristics of high software design performers and how they conduct design. High performers typically spent more time on feedback processing and less time on task-irrelevant cognitions. High performers produced more solution visualizations as helpful cognitive tools. High performers verbalized fewer task-irrelevant cognitions than moderate performers. There was only partial support for the hypothesis that high performers spend more time on planning. High performers did not spend more time on problem comprehension early in the process.

6. DMBehavior Paper Summary - Eleven papers were classified into five DMBehavior sub-classes. These five sub-classes dealt with different aspects of human thinking and behaviors. Decision makers typically do not seek optimal results through thorough reasoning and argumentation. Instead they often use naturalistic decision making approach and they are satisficed with sub-optimal solutions. Software architects face cognitive difficulties and limitations in handling highly complex problems. They also suffer from cognitive biases. Comparing with novices, experienced designers are better in exploring problem spaces and they use feedback to guide them design. There are some hints on how experts better explore the problem space and use a more efficient decision making strategy. We have not found any research that deals with decision making from a behavioral science perspective.

B. *Decision Making Practice*

We found twenty-two research papers about decision making processes, methods or tools. All of these papers study some aspects of software architecture decision making practices. We call them **DMPractice** papers. We found six sub-classes. *Decision making process* contains papers that investigate the steps software architects take in decision making. *Decision making methods* investigated how a particular method improves decision making. A specific area of group decision making is *agile development* method. Agile development methods prescribe steps to facilitate a group of developers to reach goals, schedules and consensus. We found papers that describe *decision making tools*. A number of papers describe how *design reasoning* can aid decision making. A number of research works focus on the role of *knowledge management* in decision making.

1. Decision Making Process - We found five decision making process papers. A decision process prescribes certain high-level steps for making design decisions. **S27** explored the design process control strategies using verbal protocol study of professional software designers. They found that designers exhibit opportunistic design behaviors (i.e. designers see a potential solution and jump to the opportunity) as well as systematic design behaviors (i.e. designers use breadth-first or depth-first exploration). The decision making process is highly iterative, with interleaved decisions between different loosely ordered levels of abstraction. **S51** described the results of a survey of software architects on their decision making process. They investigated the decision making scope, decision classification and the level of decisions. They found that locally scoped decisions such as a component are typically made by individuals but architectural decisions are made by a team. They also found that previous decisions, product life cycle, user requirements, time and personal preferences influence decision making. **S61** investigated how technology solutions are being considered by architects during the design process, and how to enhance architectural knowledge management to support technology decision making. **S73** presented a survey of the difficulties for making architectural design decisions. Architects consider two to three quality attributes in an architecture decision. The inter-dependencies with other decisions contribute much to the difficulty of decisions. They also found that, generally, good decisions considered more alternative solutions than bad decisions. **S26** is a study of interviewing twenty-five system analysts, team leads and senior developers to understand decision making in organization. They found eight factors that influence decision making: company size; business factors; organizational factors; technical factors; cultural factors; individual factors; project factors; and decision scope.

2. Decision Making Methods - The use of decision making methods started in the 1980s. [25]. Four papers were found in this sub-class. **S32** suggested using the *descriptive forces viewpoint* for architectural decisions. The study used 3 case studies with students to show that decisions and forces had to be documented explicitly, which caused the students to think more concretely about available decision alternatives. All groups using the forces views triggered them to consider quality attribute requirements. **S33** investigated whether junior software designers benefit from support for rational architectural decisions by the *decision viewpoint* concept. It was found that the decision viewpoint supported identification of architectural significant requirements (ASR), requirement negotiation, requirement prioritization, discovery of design option and combination of options, tradeoff analysis, validation of options against ASR, and architecture evaluation. **S39** studied the influence of *risk checklists* and the roles on risk perception and decision-making of software practitioners. It was found the practitioners who used the risk checklist identified significantly more risks than subjects who did not use it. **S49** analyzed the result of applying *Question, Option and Criteria (QOC)*. It was found that QOC helped expose assumptions, raised new questions, challenged criteria, and pointed to ways in which new options can capitalize on the strengths and overcome the weaknesses of current options. In this study, it was noted that there was a strong tendency for designers to look for evidence to confirm their initial biases.

3. Agile Software Development Method - Cockburn and Highsmith eloquently frame Agile Software Development (ASD) as people centric [26]. Decision making in agile development is one basic aspect of ASD. We found two ASD decision making related research papers. **S14** and **S15** are from the same authors. They conducted a study involving 43 practitioners in a focus group study. They found a number of decision making issues in ASD: 1) team members rely on Scrum master to commit to a decision, and decisions lack commitment; 2) information is not collected rationally and conflicting priorities exist; 3) behaviors are adapted to group dynamics and team composition is unstable; 4) team members sometimes are uncertain about who should make decisions and they rely on others to make decisions. This behavior affects decision ownership and commitment; 5) collaborative decision making prevents experts from making decisions resulting in lack of empowerment.

4. Decision Making Tools - We found three decision making tool papers. **S9** presented *gIBIS* as a hypertext tool, together with *itIBIS*. Using a case study, the researchers compared and reviewed how design rationale might make design decision making more rigorous and error free. The tool facilitated communication between team members because the underlying knowledge helped teams to detect when a conversation had wandered. A graphical representation helped participants to understand the complex issues and devise new solutions. Researchers also identified issues with the tool, such as scalability issues due to capturing knowledge, and the lack of motivation of a designer to capture knowledge used by others. **S21** proposed a meta-model to capture decision making constraints with defined semantics and a collaborative architecture decision making approach. The researchers conducted a controlled experiment of the approach and the tool (*CoCoADviSE*) involving 48 people. They found that automatic enforcement of constraints increases the effectiveness and efficiency of decision making because it takes away the burden of detecting, preventing and resolving constraint violations "manually" from the user. **S46** reported two other experiments on CoCoADviSE using students. These experiments found that students needed less time to document design decision using the tool.

5. Design Reasoning - Design reasoning is a process that makes use of information and design rationale to support logical argumentation in decision making. Design rationale is the justifications of a decision. [27] reported that well-structured design rationale is a documentation that helps designers track and evaluate the issues and alternatives being explored. Many design rationale methods have been suggested [28, 29]. Though design rationale provides design justifications, these methods do not show how the process of reasoning is carried out. We identified four papers that deal with design reasoning.

S30 was a survey of fifty-three professionals to find out how they reason in real projects. Software architects often searched for multiple design options when making decisions, they consider interconnected decisions. They usually think about the pros and cons of design options but they seldom reject decisions they made before. **S31** was a survey of undergraduate students about naïve reasoning for architecture decision making. Students were taught to consider the ASRs and put emphasis on the quality attribute requirements. However, many students did not identify the most challenging requirements, nor did they prioritize them. Students did not relax requirements to yield more design options and they also declared that they preferred well known solutions in favor of unknown alternatives. Also, they did not seem to be aware of limitations and constraints that the solutions impose on other decisions. Students weighted pros and cons of design options, but they did not consciously make trade-offs between requirements, and they neglected to validate the decisions against each other. Students did not seem to be aware of the dependencies and the relationships between architectural decisions. The students quickly came up with a first architectural vision and did not significantly deviate from this vision any more. This is another indicator that students did not critically evaluate their decisions. **S67** was a survey on how practitioners think about and reason about design decision and design rationale practices. It was found that the following design rationales were used to support decision making: constraints, assumptions, weakness, cost, benefits, complexity, certainty of design, certainty of implementation and tradeoff. Additionally, design rationales that positively justify a design receive more attention than those negative rationales that explain why the design may have issues. That leads the researchers to suspect that there might be a tendency or a bias towards presenting "good news" rather than "bad news". **S69** explored the effects of design reasoning on the quality of design by comparing two groups of practitioners in a controlled experiment environment. It was found that for junior designers, explicitly stating their design rationale helped improve design quality. By explicitly stating design issues and options, the test group performed more systematic design reasoning and was able to back track their decisions.

6. Knowledge Management - Experience and knowledge play a role in decision making and management of such knowledge can facilitate decision support. Knowledge management encompasses knowledge capture (in terms of documentation), sharing and communication [4]. Four papers were found in this area. **S47** identified patterns for service-based integration based on a systematic literature review. The identified patterns are grouped into four decision levels: architecture, platform, integration and application. **S35** investigated knowledge sharing between software architects and characterizes their position as decision maker. It was found that architects spent most of their time on making architectural decisions and less time on documenting the decision results. **S77** presented an interview-based case study of practitioners about design decisions and their documentation. In documenting architecture decisions, architects classified design decisions according to granularity, scope and impact. Low-level decisions with a local scope are often called design decisions or implementation decisions, whereas high-level decisions with a global scope are typically referred to as architectural decisions. **S19** introduced valued-based documentation of design rationale. Researchers identified useful design decision information such as issue articulation, design decisions, requirements, position and alternatives, arguments, constraints, assumptions, related decisions, status, related principles, artifacts and notes. This knowledge serves future decisions.

7. DMPractice Paper Summary - A number of observations arise from the study of decision making practice papers. First, a number of papers have identified that decision complexity is one of the main issues. To remedy this problem, they propose methods such as capturing decision chains and visualization tools to help software developers. Second, designing and decision making do not follow a prescribed process. Architects are opportunistic and their design focus shifts as they move through design problems and solution spaces. Third, architects can fixate on interim decisions that have been made, and do not change their decisions despite arrival of new contradicting information. Fourth, when working together with a method like ASD, issues such as decision deference and indecision can arise. Whilst most DMPractice papers have proposed some process, methods or tools to aid decision making, almost all the papers in this class do not discuss the contributing human behavioral issues.

IV. DISCUSSIONS

Based on the literature review, we make observations on the overall research scene in software architecture decision making; we discuss what we have learned, and we identify new research opportunities.

A. Few Human Aspect Research Works

Lenberg et al. suggest that even though the software engineering field recognizes the importance of human aspects, the main research focus has been on technology [11]. We found similar phenomena here. Out of the thousands of papers published in the selected venues over 11 years, we only found thirty-three papers on software architecture decision making that satisfy our selection criteria. This is a smaller number. In these thirty-three papers, only eleven papers deal with human behaviors, whereas twenty-two papers studied process, methods, techniques or tools. It seems to indicate that the emphasis is on methods and tools rather than decision behaviors. We argue that decision making practices are rooted in the way software architects think and act. *In order to improve software architecture decision making practice, it is necessary to carry out more studies on decision making behaviors*.

B. Symbiotic Relations of Behavioral and Practice Research

A software project is typically built by many people having differing personalities and differing skills, working in a physical environment within an organizational culture [26]. **S10** found that many managerial decisions are unconscious with many cognitive biases. If we are unaware of these unconscious cognitive activities of decision makers, we may not realize that these issues could adversely influence the execution of a development method. DMBehavior works *provide fundamental and important knowledge* that underpins software decision making tools and techniques. DMPractice papers *focus on software development practices, methods and tools that improve decision making behavior*. Observations of subjects in situ of a decision process provide insights on how software developers work in their specific environments and context. Fig. 2 illustrates a symbiotic relationship. *Both research areas are necessary to provide a complete picture to improve software architecture decision making.* For instance, **S32** and **S33** propose different viewpoints to shape decision making. These methods describe how software developers can design better in a certain context. Software developers may use such a decision making method to overcome human issues such as cognitive limitations through better focus and tool support. Both research areas can mutually benefit by leaning on each other. There are many research opportunities to further investigate behavioral decision making in software practice. For instance, it would be interesting to see how improvements within design reasoning or knowledge management impact on cognitive biases, cognitive limitations and satisficing behavior. Also, it would be interesting to understand how to measure or identify the extent of cognitive biases and limitations in software architecture decision making.

C. Behavioral Software Architecture Decision Making Research Topics

From the literature review, we have noticed that many human behavioral issues have not been attended by software architecture researchers. These issues are fundamental to formulating process, methods and tools to aid architects in their practices. We summarize them into three research topics.

Decision Making Heuristics. Software design complexity increases as requirements become more interrelated, technologies become more advanced and the needs of customers grow and diversify. Software architects naturally employ decision making heuristics in such an environment. It was found that 50%-70% of management decisions are unconscious in general. Anchoring-adjustment, availability heuristic, representative heuristic and moral judgments all play a role in decision making [30]. For instance, an architect may choose to explore one particular potential solution and then grows to love it, subsequently ignoring other potentially good solutions (i.e. anchoring and not changing).

It has been suggested that intuitive (unconscious, system 1) and rational (conscious, system 2) processes complement each other in decision making [17, 31]. In this review, a number of papers such as **S80** show that software professionals often use naturalistic decision making and sometimes rational decision making. The choice of decision heuristics is often implicit, but it influences decision outcomes. In [30], Crowder listed decision heuristics used by senior managers, and each heuristic comes with potential biases. We tabulate some of these decision heuristics and our interpretation of software biases in Table II. The list of decision heuristics in Table II is likely not all the heuristics there are. We need to identify them and learn more about them. At this stage, we understand that decision heuristics and decision making behavior naturally occur. How they are used by architects can produce different results, some better and some worse. Our question is: *What decision making heuristics can software architects use to cope with architecture design complexity?* Currently we know little about this area. Further studies of decision making heuristics, the potential issues and counter measures can be beneficial in providing decision making mental tools.

TABLE II. DECISION MAKING HEURISTICS AND POTENTIAL COGNITIVE BIAS (ADAPTED TO SOFTWARE ARCHITECTURE)

Decision Heuristics	Potential Biases in Software Development
Anchor / Adjustment	Designers fixate on an initial software architecture design and unwilling to consider a better alternative
Availability	Designers make decision on what heuristics/knowledge is immediately known to him, instead of exploring unknown solutions
Representative	Judging a preconceived scenario as representative of a general situation. Designers sometimes guess whether a certain use case scenario is a general scenario and how often it happens, then design software to cater for that.
Moral	Designers make decision based on what one thinks is right. They judge whether they should design software that benefits end user or the company they work for, especially if the goals contradict.
Elimination by Aspect	Decision makers focus on one aspect and eliminate alternatives that do not have this aspect. Software developers may eliminate a design for performance if the prime focus is security.

Mental Representation and Limitations. Langley et al. suggest that decision making is not as structured as some researchers theorize, decision making can be “dark and tangled” [32]. Parnas and Clements suggest that software developers do not always design systematically. They fake rationality by providing design rationale after the fact [33]. Guindon et al. find that software designers do not follow a structured design process. They observe that designers are opportunistic and can veer off to areas that they are most interested in at the time [34]. Björklund suggests that experts use mental representation to associate information and tackle problems [24]. Software developer’s mental capacity is limited by memory capacity and processing power and bounded rationality [9]. All these works point to some human limitations. **S28** describes the difficulties on handling design complexity, whilst **S70** describes the satisficing behaviors of developers. In order to overcome some of these human limitations, **S7** suggests that a creative process of exploration, hypotheses testing and problem recognitions are important. **S62** discuss feedback processing, solution visualization and task-irrelevant cognitions. From these discussions, a number of research questions are worth asking: *What steps do we take to improve mental capabilities for better software architecture design? How do we check cognitive limitations? Can we create better models and tools for software architects based on the understanding of expert mental representation?*

De-biasing. Cognitive biases have been found to play a role in general decision making [35]. **S53** shows the presence of framing bias in requirements elicitation. Framing bias was also shown in medical decision making [36]. In system development, stakeholders were shown to be biased in many ways [19]. Stacy and Macmillian gave anecdotal examples of cognitive biases, on inheritance and dynamic binding, in software engineering [37]. Vliet and Tang gave anecdotal examples of cognitive biases in [6]. Thus, the question is: *How do we de-bias and reduce or eliminate the effect of biases?* Keren's framework for de-biasing in medical diagnosis and prescription is an example for reference [38]. In summary, we need to *(i) understand the environment that creates biases (ii) study and apply alternative means for reducing or eliminating biases (iii) monitor and evaluate the effectiveness of de-biasing technique(s).* Kahneman [35] suggests to use a checklist to reveal fundamental decision making thoughts: (a) is there something to suspect motivated errors or errors driven by the self-interest of the recommending team? (b) have the people making the recommendation fallen in love with it? (c) were there dissenting opinions within the recommending team? Reflection might also be another means to encourage reasoning and check biases. Asking simple questions seems to have an effect on stipulating design reasoning [39]. Other works in software architecture reviews use a rational and systematic approach, based on decision information and rationale to review design decisions [40, 41].

V. CONCLUSION

Decision making is a unique human activity involving many aspects such as cognition, behaviors and group interactions. In software architecture decision making research, researchers have investigated both behavior and practice aspects of this activity. The factors that influence decision making are complex and intertwining. We wanted to understand the current state of research on software architecture decision making, in terms of human behaviors and practice and how they are related to each other. We also wanted to understand what further research questions we can ask. To achieve these goals, we conducted a literature review of eight different research publication venues, between 2005 and 2015, to search for empirical papers on human aspects in decision making. We classified these papers into two major classifications, decision making behaviors and decision making practice. To aid our analysis, we referenced decision making research works from other disciplines to give us some context.

Our main conclusions are, firstly, there are few research works on human aspects in software architecture decision making. We only found 33 papers and there is an apparent lack of knowledge to improve decision making practices. Second, there exists a symbiotic research relationship between decision making behavior and decision making practice. Knowledge from decision making behavior can underpin practice improvements, and knowledge from decision making practice can improve our understanding of decision behavior. Third, three research topics are identified, (a) formulating decision making heuristics to cope with design complexity; (b) providing aids to assist the mental capabilities of software architects to cope with cognitive limitations; and (c) dealing with cognitive biases. For the future, a systematic literature review with a wider scope that includes other software engineering venues such as CHASE workshop and searches beyond the past decade would be valuable.

REFERENCES

[1] T. Moran and J. Carroll, *Design Rationale: Concepts, Techniques, and Use*: Lawrence Erlbaum Associates, Publishers, 1996.

[2] J. Tyree and A. Akerman, "Architecture Decisions: Demystifying Architecture," *IEEE Software,* vol. 22, pp. 19-27, 2005.

[3] J. Burge and D. C. Brown, "Reasoning with design rationale," in *Artificial Intelligence in Design'00*, ed: Springer, 2000, pp. 611-629.

[4] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, and M. A. Babar, "10 years of Software Architecture Knowledge Management: Practice and Future," *Journal of Systems and Software,* 2015.

[5] M. A. Babar, T. Dingsøyr, P. Lago, and H. v. Vliet, Eds., *Software Architecture Knowledge Management: Theory and Practice*. Springer-Verlag Berlin Heidelberg, 2009, p.^pp. Pages.

[6] H. van Vliet and A. Tang, "Decision making in software architecture," *Journal of Systems and Software,* vol. 117, pp. 638-644, 2016.

[7] P. Kruchten, "What do software architects really do? ," *Journal of Systems and Software,* vol. 81, pp. 2413-2416, 2008.

[8] D. Kahneman, "Maps of bounded rationality: Psychology for behavioral economics," *The American economic review,* vol. 93, pp. 1449-1475, 2003.

[9] H. Simon, *Models of Bounded Rationality*: MIT Press Cambridge, Mass, 1982.

[10] J. Carroll and E. Johnson, *Decision research : a field guide* Sage, 1990.

[11] P. Lenberg, R. Feldt, and L. G. Wallgren, "Behavioral software engineering: A definition and systematic literature review," *Journal of Systems and Software,* vol. 107, pp. 15-37, 2015.

[12] M. Galster and D. Weyns, "Empirical, Research in Software Architecture-How far have we come?," in *Proceedings of the 13th Working IEEE/IFIP Conference on Software Architecture*, 2016, pp. 1-8.

[13] M. Petre, A. van der Hoek, and A. Baker, "Editorial," *Design Studies,* vol. 31, pp. 533-544, 2010.

[14] A. Jansen and J. Bosch, "Software Architecture as a Set of Architectural Design Decisions," presented at the Proceedings 5th IEEE/IFIP Working Conference on Software Architecture, 2005.

[15] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 2014, p. 38.

[16] D. Lovallo and O. Sibony, "The case for behavioral strategy," *McKinsey Quarterly,* vol. 2, pp. 30-43, 2010.

[17] D. Kahneman, *Thinking, fast and slow*: Penguin, 2011.

[18] D. Kahneman, A. Tversky, and P. Slovic, *Judgment under uncertainty heuristics and biases*. New York: Cambridge University Press, 1982.

[19] D. Arnott, "Cognitive biases and decision support systems development: a design science approach," *Information Systems Journal,* vol. 16, pp. 55-78, 2006.

[20] R. Hastie and T. Kameda, "The robust beauty of majority rules in group decisions," *Psychological review,* vol. 112, p. 494, 2005.

[21] R. Guindon, H. Krasner, and B. Curtis, "Breakdowns and processes during the early activities of software design by professionals," in *Empirical studies of programmers: second workshop*, M. O. Gary, S. Sylvia, and S. Elliot, Eds., ed: Ablex Publishing Corp., 1987, pp. 65-82.

[22] H. A. Simon, *The Sciences of the Artificial*, 3rd ed.: The MIT Press, 1996.

[23] H. A. Simon, "Satisficing," *The new Palgrave: a dictionary of economics,* vol. 4, pp. 243-245, 1987.

[24] T. A. Björklund, "Initial mental representations of design problems: Differences between experts and novices," *Design Studies,* vol. 34, pp. 135-160, 2013.

[25] J. Lee, "Design Rationale Systems: Understanding the Issues," *IEEE Expert,* vol. 12, pp. 78-85, 1997.

[26] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer,* vol. 34, pp. 131-133, 2001.

[27] J. Lee and K. Lai, "What is Design Rationale?," in *Design Rationale - Concepts, Techniques, and Use*, T. Moran and J. Carroll, Eds., ed New Jersey: Lawrence Erlbaum, 1996, pp. 21-51.

[28] A. Tang, M. A. Babar, I. Gorton, and J. Han, "A survey of the use and documentation of architecture design rationale," in *Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on*, 2005, pp. 89-98.

[29] D. Falessi, G. Cantone, R. Kazman, and P. Kruchten, "Decision-making techniques for software architecture design: A comparative survey," *ACM Computing Surveys (CSUR),* vol. 43, p. 33, 2011.

[30] M. Crowder, "Decision-making in practice: The use of cognitive heuristics by senior managers," PhD, University of Chester, 2013.

[31] G. Klein, *Streetlights and shadows: Searching for the keys to adaptive decision making*: The MIT Press, 2009.

[32] A. Langley, H. Mintzberg, P. Pitcher, E. Posada, and J. Saint-Macary, "Opening up decision making: The view from the black stool," *organization Science,* vol. 6, pp. 260-279, 1995.

[33] D. Parnas and P. Clements, "A Rational Design Process: How and Why to Fake It," *IEEE Transactions on Software Engineering,* vol. 12, pp. 251-257, 1985.

[34] R. Guindon, "Knowledge Exploited by Experts During Software Design," *International Journal of Man-Machine Studies,* vol. 33, pp. 279-304, 1990.

[35] D. Kahneman, D. Lovallo, and O. Sibony, "Before you make that big decision," *Harvard Business Review,* vol. 89, pp. 50-60, 2011.

[36] J. Gong, Y. Zhang, Z. Yang, Y. Huang, J. Feng, and W. Zhang, "The framing effect in medical decision-making: a review of the

literature," *Psychology, health & medicine,* vol. 18, pp. 645-653, 2013.
[37] W. Stacy and J. MacMillan, "Cognitive bias in software engineering," *Commun. ACM,* vol. 38, pp. 57-63, 1995.
[38] G. Keren, "Cognitive aids and debiasing methods: can cognitive pills cure cognitive ills?," *Advances in psychology,* vol. 68, pp. 523-552, 1990.
[39] M. Razavian, A. Tang, R. Capilla, and P. Lago, "In two minds: how reflections influence software design thinking," *Journal of Software: Evolution and Process,* vol. 28, pp. 394-426, 2016.
[40] U. van Heesch, V. P. Eloranta, P. Avgeriou, K. Koskimies, and N. Harrison, "Decision-Centric Architecture Reviews," *Software, IEEE,* vol. 31, pp. 69-76, 2014.
[41] A. Tang and M. F. Lau, "Software architecture review by association," *Journal of Systems and Software,* vol. 88, pp. 87-101, 2// 2014.

SELECTION OF DECISION MAKING LITERATURE

[S3] K. Børte, S. R. Ludvigsen, and A. I. Mørch, "The role of social interaction in software effort estimation: Unpacking the "magic step" between reasoning and decision-making," Information and Software Technology, vol. 54, pp. 985-996, 2012.
[S7] H. Christiaans and R. A. Almendra, "Accessing decision-making in software design," Design Studies, vol. 31, pp. 641-662.
[S9] E. Conklin and K. C. Burgess-Yakemovic, "A Process-Oriented Approach to Design Rationale," in Design Rationale: Concepts, Techniques and Use, T. Moran and J. Carroll, Eds., ed: Lawrence Erlbaum Associates, 1996, pp. 393-427.
[S14] M. Drury, K. Conboy, and K. Power, "Decision making in agile development: A focus group study of decisions and obstacles," in Agile Conference (AGILE), 2011, 2011, pp. 39-47.
[S15] M. Drury, K. Conboy, and K. Power, "Obstacles to decision making in Agile software development teams," Journal of Systems and Software, vol. 85, pp. 1239-1254, 2012.
[S19] D. Falessi, G. Cantone, and P. Kruchten, "Value-based design decision rationale documentation: Principles and empirical feasibility study," in Software Architecture, 2008. WICSA 2008. Seventh Working IEEE/IFIP Conference on, 2008, pp. 189-198.
[S21] P. Gaubatz, I. Lytra, and U. Zdun, "Automatic enforcement of constraints in real-time collaborative architectural decision making," Journal of Systems and Software, vol. 103, pp. 128-149, 2015.
[S26] I. Groher and R. Weinreich, "A Study on Architectural Decision-Making in Context," in Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on, 2015, pp. 11-20.
[S27] R. Guindon and B. Curtis, "Control of cognitive processes during software design: what tools are needed?," presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, Washington, D.C., United States, 1988.
[S28] R. Guindon, H. Krasner, and B. Curtis, "Breakdowns and processes during the early activities of software design by professionals," in Empirical studies of programmers: second workshop, M. O. Gary, S. Sylvia, and S. Elliot, Eds., ed: Ablex Publishing Corp., 1987, pp. 65-82.
[S30] U. van Heesch and P. Avgeriou, "Mature architecting-a survey about the reasoning process of professional architects," in Software Architecture (WICSA), 2011 9th Working IEEE/IFIP Conference on, 2011, pp. 260-269.
[S31] U. van Heesch and P. Avgeriou, "Naive Architecting-Understanding the Reasoning Process of Students," in Software Architecture, ed: Springer, 2010, pp. 24-37.
[S32] U. Van Heesch, P. Avgeriou, and R. Hilliard, "Forces on architecture decisions-a viewpoint," in Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on, 2012, pp. 101-110.
[S33] U. van Heesch, P. Avgeriou, and A. Tang, "Does decision documentation help junior designers rationalize their decisions? A comparative multiple-case study," Journal of Systems and Software, vol. 86, pp. 1545-1565, 2013.
[S35] J. F. Hoorn, R. Farenhorst, P. Lago, and H. Van Vliet, "The lonesome architect," Journal of Systems and Software, vol. 84, pp. 1424-1435, 2011.
[S39] M. Keil, L. Li, L. Mathiassen, and G. Zheng, "The influence of checklists and roles on software practitioner risk perception and decision-making," Journal of Systems and Software, vol. 81, pp. 908-919, 2008.
[S46] I. Lytra, P. Gaubatz, and U. Zdun, "Two controlled experiments on model-based architectural decision making," Information and Software Technology, vol. 63, pp. 58-75, 2015.
[S47] I. Lytra, S. Sobernig, and U. Zdun, "Architectural decision making for service-based platform integration: A qualitative multi-method study," in Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on, 2012, pp. 111-120.
[S49] A. MacLean, R. M. Young, V. M. Bellotti, and T. P. Moran, "Questions, options, and criteria: Elements of design space analysis," Human–computer interaction, vol. 6, pp. 201-250, 1991.
[S51] C. Miesbauer and R. Weinreich, "Classification of Design Decisions–An Expert Survey in Practice," in Software Architecture, ed: Springer, 2013, pp. 130-145.
[S52] N. B. Moe, A. Aurum, and T. Dybå, "Challenges of shared decision-making: A multiple case study of agile software development," Information and Software Technology, vol. 54, pp. 853-865, 2012.
[S53] R. Mohanani, P. Ralph, and B. Shreeve, "Requirements fixation," in Proceedings of the 36th International Conference on Software Engineering, 2014, pp. 895-906.
[S60] V. Smrithi Rekha and H. Muccini, "A study on group decision-making in software architecture," in Software Architecture (WICSA), 2014 IEEE/IFIP Conference on, 2014, pp. 185-194.
[S61] M. Soliman, M. Riebisch, and U. Zdun, "Enriching Architecture Knowledge with Technology Design Decisions," in Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on, 2015, pp. 135-144.
[S62] S. Sonnentag, "Expertise in Professional Software Design: A Process Study," Journal of Applied Psychology, vol. 83, pp. 703-715, 1998.
[S66] A. Tang, A. Aleti, J. Burge, and H. van Vliet, "What makes software design effective?," Design Studies, vol. 31, pp. 614-640, 2010.
[S67] A. Tang, M. A. Babar, I. Gorton, and J. Han, "A survey of the use and documentation of architecture design rationale," in Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on, 2005, pp. 89-98.
[S69] A. Tang, M. H. Tran, J. Han, and H. van Vliet, "Design Reasoning Improves Software Design Quality," in Proceedings of the Quality of Software-Architectures (QoSA 2008), 2008, pp. 28-42.
[S70] A. Tang and H. van Vliet, "Software Designers Satisfice," in Software Architecture. vol. 9278, D. Weyns, R. Mirandola, and I. Crnkovic, Eds., ed: Springer International Publishing, 2015, pp. 105-120.
[S73] D. Tofan, M. Galster, and P. Avgeriou, "Difficulty of architectural decisions–a survey with professional architects," in European Conference on Software Architecture, 2013, pp. 192-199.
[S77] R. Weinreich, I. Groher, and C. Miesbauer, "An expert survey on kinds, influence factors and documentation of design decisions in practice," Future Generation Computer Systems, vol. 47, pp. 145-160, 2015.
[S80] C. Zannier, M. Chiasson, and F. Maurer, "A model of design decision making based on empirical results of interviews with software designers," Information and Software Technology, vol. 49, pp. 637-653, 2007.
[S81] C. Zannier and F. Maurer, "Social Factors Relevant to Capturing Design Decisions," in Proceedings of the Second Workshop on SHAring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent, 2007, p. 1.