

# A Qualitative Empirical Evaluation of Design Decisions

Carmen Zannier  
University of Calgary  
2500 University Drive NW  
Calgary, AB, Canada  
011 403 210 8153

zannierc@cpsc.ucalgary.ca

Frank Maurer  
University of Calgary  
2500 University Drive NW  
Calgary, AB, Canada  
011 403 220 3531

maurer@cpsc.ucalgary.ca

## ABSTRACT

In this paper, we motivate examining software design decision making and provide the process by which the examination will occur. The objective is to provide qualitative results indicative of rational or naturalistic software design decision making. In a rational decision a decision maker evaluates decision alternatives and potential outcomes for each alternative using a utility function and probabilities of the outcome of each alternative. The utility function assigns a value to each possible alternative based on its outcome. The goal of rational decision making is selecting the optimal alternative. A naturalistic decision manifests itself in dynamic and continually changing conditions, embodies real-time reactions to these changes, embraces ill-defined tasks, and has a goal of selecting a satisfactory alternative. The proposed empirical qualitative study consists of inductive and deductive interviewing and deductive observations.

## Categories and Subject Descriptors

K.6.1 [Project and People Management]: Systems. Analysis and Design

## General Terms

Design, Human Factors.

## Keywords

Rational Decision, Naturalistic Decision, Qualitative, Empirical, Interviews, Observations,

## 1. INTRODUCTION

Examining software design decision making is an important topic of study because of the novelty of the topic, the utility for software design and the complexity involved in the topic. The objective is to provide qualitative results indicative of rational or naturalistic software design decision making. The research question addressed is this: How do people make software design decisions?

Firstly, little work exists concerning how decisions are made in software design, [8],[17], but much evidence shows ubiquitous design decisions in software development and significant impacts of design decisions on software development [41],[16],[17]. Little empirical work exists on high-level software design itself, where main functions, information flow, interfaces between functions and conceptual data structures are defined [13],[14]. Support for examining software design in general and specifically software design decisions, is in abundance [1][8][16][17][41][51].

Secondly, this is a useful topic of study for software design because it provides insight into the human behaviour surrounding software design. "The major problems of [software design] work are not so much technological as sociological in nature" [9].

The final motivation behind examining software design decision making is the structural challenge involved in the examination. The research process proposed is a qualitative ethnomethodological study. Qualitative studies emerge from interviews, observations and written documents [36]. The use of a qualitative study of software design introduces a challenge of structure in two ways.

Firstly, the credibility of a qualitative study can be questioned from a traditional scientific research perspective via points such as objectivity of the inquirer, validity of the data, rigour of fieldwork procedures, and reliability of pattern analysis [36].

Secondly, software design is currently defined as a "wicked" problem [40],[39]. A wicked problem has no definitive formulation, no stopping rule, no right or wrong, and no immediate or ultimate test of a solution.

The potential subjectivity and ambiguity in both qualitative studies and software design make this topic complex. The novelty of studying software design decision making, the utility of examining human behaviour in software design, and the complexity required in defining the examination of software design decision making, make this topic an important and interesting topic of study.

Section 2 provides background information on rational decision making, naturalistic decision making and software design studies. Section 3 presents the details of the study. Section 4 presents the current state of the study and Section 5 concludes this work.

## 2. BACKGROUND

We provide a discussion of software design as an ill-structured problem, decision making, design rationale, software design studies and qualitative studies.

### 2.1 Software Design

Software design is a problem structuring activity accomplished throughout the software development lifecycle.

#### 2.1.1 Problem Structuring

An ill-structured problem (ISP) is a problem that is not well-structured [46]. A well-structured problem (WSP) is a problem that has criteria that reveal relationships between the characteristics of a problem domain and the characteristics of a method by which to solve the problem [46]. The majority of problem solving is problem structuring, through which a problem

solver converts an ISP to a WSP [46]. A problem solver spends more time in problem structuring than in actually solving a problem once it is structured [46]. An example of an ISP is the design of a new house [46]. Decisions made in early design sketches of the house establish structures under which following decisions will be made [46].

### 2.1.2 Problem Structuring & Wicked Problems

Evidence of software design as a problem structuring activity is in abundance [12],[13],[14],[15], and software design as a wicked problem is also ubiquitous [40],[39]. The first property of a wicked problem states that specifying the problem directs the solution [40]. This definition aligns with that of problem structuring. For the purposes of this work, the concept of design as an ISP or as a wicked problem is considered one and the same.

### 2.1.3 Software Design & Decision Definitions

We identify five abstraction levels at which software design occurs. Table 1 lists these abstraction levels and their definitions.

**Table 1: Software Design Tiers**

Implementation Design	
	Activities related to lexical, or syntactical issues of a software application. Implementation design addresses the question: How are we going to code this software application?
Structural Design	
	Activities related to the definition and allocation of application domain roles and application domain responsibilities. Structural Design addresses the question: What entity does what task?
Interaction Design	
	Activities related to the communication among entities defined in structural design. Interaction design addresses the question: Which entities communicate with each other?
Architectural Design	
	Activities related to deployment issues of a software application. Architectural design addresses the question: How are we going to implement this software application. It differs from Implementation Design in that the implementation constructs considered are of a much broader scope.
Usability Design	
	Activities related to the support of application users' needs from a user interface and work task perspective. Usability design addresses the question: How will people use the system? [6]

Software design is performed at five tiers and design decisions are made at five corresponding tiers. Table 2 provides definitions of a design decision and decisions at the five design tiers.

In view of the definition of design as a problem structuring activity, there is little support for the belief that these design tiers are mutually exclusive or that the corresponding design decisions are mutually exclusive.

**Table 2: Design Decision Definitions and Definition Tiers**

Design Decision	
	A selection of an option among zero or more known and unknown options concerning the Implementation, Structure, Interaction and Usability of a software application.
Implementation Decision	
	A selection of an option among zero or more known and unknown options concerning activities related to lexical, or syntactical issues of the software application.
Structural Decision	
	A selection of an option among zero or more known and unknown options concerning activities related to the allocation of application domain roles and application domain responsibilities to programmable entities.
Interaction Decision	
	A selection of an option among zero or more known and unknown options concerning activities related to the communication among entities defined in Structural Design.
Architectural Decision	
	A selection of an option among zero or more known and unknown options concerning activities related to deployment issues of the software application.
Usability Decision	
	A selection of an option among zero or more known and unknown options concerning activities related to the support of application users' needs from a user interface and work task perspective.

## 2.2 Decision Making

We present two decision making areas each with a definition of a decision and each with decision making approaches. A rational decision "is one that conforms either to a set of general principles that govern preferences or to a set of rules that govern behaviour. These principles or rules are then applied in a logical way to the situation of concern resulting in actions which generate consequences that are deemed to be acceptable to the decision maker" [48]. A naturalistic decision "connotes situational behaviour without the conscious analytical division of situations into parts and evaluation according to context-independent rules" [11],[25].

### 2.2.1 Rational Decision Making

The rational decision making domain is characterized by an appreciation for mathematical computation of decision alternatives. A rational decision is also known as a normative decision and strives for an optimal selection among alternatives.

A rational decision has three features that are parts of the decision maker's approach to decision making [45],[44]. Firstly, decision alternatives are represented by a set of possible courses of action and potential outcomes for each action. Secondly, a utility function assigns a value to each possible action based on the attributes of its outcome. Lastly, a decision has information or probabilities as to which outcome will occur in case of the selection of a particular alternative.

Rational decision making is limited by three assumptions. The first is that a set of possible courses of action and the probability of outcomes is known. The second is that the decision maker's goal is to optimize. The last assumption is that combinatorial explosion of alternatives and the time involved in mathematical calculations of situations of combinatorial explosion are not a large concern [35],[27].

The Analytical Hierarchy Process (AHP) is a rational decision theory process in which a decision maker specifies all possible outcomes and ranks them according to objective criteria, external to the decision maker's value system [42].

### 2.2.2 Naturalistic Decision Making

The naturalistic decision making domain is characterized by an appreciation for real-time scenarios and expert judgment biases. A naturalistic decision is defined by six characteristics [26].

Firstly, a naturalistic decision manifests itself in dynamic and continually changing conditions. It embodies fast reactions to these changes. It embraces ill-defined tasks and goals. A naturalistic decision resolves itself under the guidance of knowledgeable decision makers. It utilizes expert situation assessment over consequential choice. Lastly, a naturalistic decision has a goal of satisficing instead of optimizing.

We define three terms mentioned in these characteristics: consequential choice, situation assessment and satisficing. Consequential Choice is the organized analysis of options and potential outcomes, typical of rational decision theory [30]. Situation Assessment is the absence of choice in real-world decisions. A decision maker exercises an action that may or may not be considered amongst a set of choices [30]. A decision maker thinks of a possible option then uses mental simulation to evaluate the option. If the mental simulation succeeds the decision maker executes that option. Lastly, Satisficing is the acceptance of a satisfactory situation [27]. Satisficing values "good enough" instead of "the best".

Naturalistic decision approaches are limited by the absence of a definable, predictable outcome.

Table 3 presents ten approaches to naturalistic decision making.

## 2.3 Design Rationale

Design rationale is the documenting of design decisions and design decision justifications [29],[25]. The purpose is to facilitate reasoning and communicating about a design as well as organizational learning [29],[25].

### 2.3.1 Design Rationale Approaches

We present two approaches as examples of the attempts to solve the issue of capturing design decisions without intruding too heavily on the design process. The first approach stems from an Issue-Based Information System (IBIS) defined by Rittel (1970). The IBIS model partitions wicked problems into Issues, Positions and Arguments. An issue is a point or concern regarding a design problem, and contains one or more positions that make a statement about an issue.

Positions contain one or more supporting or refuting arguments [7]. Extensions of IBIS are in abundance [7],[29],[10]. The second approach to modelling design rationale amalgamates Questions, Options and Criteria (QOC) [31]. A decision

**Table 3: Naturalistic Decision Approaches**

Requisite Decision Model	
	A shared social reality represents a decision problem and is requisite if the participants of the shared social reality deem the form and content of the model to be sufficient to solve a problem [38].
Prospect Theory	
	Introduces the certainty effect, which states that a decision maker will select more certain situations when possible gain is large, even if the certain situation results in a smaller gain than the alternative [19].
Elimination by Aspects	
	Examines uncertainty through probabilistic elimination of alternatives via subjective aspects. [49].
Social Judgment Theory	
	Defines a Cognitive Continuum Index which fixates on the conflict between intuitive and analytical cognition, converging on quasi-rationality [4].
Heuristics and Biases	
	Examines three factors affecting decision making: representativeness, availability, and anchoring & adjustment [50].
Dominance Structuring	
	Dominance Structuring uses bolstering and deemphasizing to select an alternative subjectively dominant above all others. [33].
Image Theory	
	Image Theory merges a forecasted plan with a trajectory view using the status quo as a guideline [3].
Explanation Based Decision Making	
	Aligns a story of an event with an alternative. [37].
Participatory Decision Making	
	Defines divergence, groaning and convergence of group decision making which values full participation, among decision participants [20].
Recognition Primed Decision Making	
	Epitomizes naturalistic decision making via situation assessment, serial evaluation, changing environment and ill defined goals and tasks [23].

alternative exists in a space of possibilities and design decisions are justified through questions, options and criteria.

External evaluations of design rationale have not been positive. The usefulness of design rationale documents and the sufficiency of capture techniques have been questioned [22],[43],[34].

### 2.3.2 Design Rationale & Rational Decision Making

We identify similarities between two rational decision making assumptions and traits of design rationale capture. Similar to the first assumption of rational decision making, design rationale also assumes that issues or options are known well enough to be recorded. Similar to the third rational decision making assumption design rationale assumes that the number of options to be

recorded will not expand beyond practical limits. The second assumption of rational decision making is that the decision maker's goal is to optimize, and in this respect, rational decision making differs from design rationale. Design rationale provides an opportunity to justify one's decision [29], and thus does not require an optimal choice.

## 2.4 Software Design Studies

A survey of software design studies, [1][2][8][12][13][14][21][32][47][51], finds that the following six related qualities impact software design: expertise, mental modelling, mental simulation, continual restructuring, preferred evaluation criteria and group interactions. Each quality is defined below, and the results of the studies pertaining to each quality, are summarized.

### 2.4.1 Design Expertise

Expertise is the knowledge and experience software designers have in design [1]. The issue of expertise, or knowledge, is fundamental to software productivity and quality and can be a scarce attribute [8]. It has been shown that individual team members do not possess enough knowledge for a project and must learn additional information before they are productive [51].

Software design studies showed that higher expertise resulted in an improved ability to create internal models, run mental simulations, work with abstract models, produce detailed models and work with more than one concept at a time [1]. Experts, collected fewer drawings and documentation than novice designers [2] and spent less time than novices on task-irrelevant cognitions. [47].

### 2.4.2 Mental Modelling

Mental modelling is the internal or external model a designer creates that is capable of supporting mental simulation [1]. "A model is an abstract representation of a system that enables us to answer questions about the system" [5]. External modelling is the development of diagrams, code or any physical model of the application [2]. Internal modelling is the mental development of a model of the application [13].

Expert designers incorporated the recognition of solutions varying in abstraction into their design problem solving [13]. The quality and detail of the mental model improved with increased expertise in the domain [1],[8],[47].

### 2.4.3 Mental Simulation

Mental simulation is the "ability to imagine people and objects consciously and to transform those people and objects through several transitions, finally picturing them in a different way than at the start" [27]. External and internal modelling provide a foundation upon which mental simulations run [12].

Mental simulations occurred throughout the software design process at varying levels of quality dependent upon the skill of the designer and the quality of the model on which the mental simulation ran [1],[8],[13]. When mental simulations based on internal models were cognitively taxing, external models formed the basis for mental simulation or the mental simulations were shallow [13].

### 2.4.4 Continual Restructuring

Continual restructuring is the process of turning an ISP to a WSP. Continual restructuring is also known as systematic expansion [1],[14].

The context of the current state of a design, design element properties and design organization determined relevant issues of the design problem [2],[32],[21]. Designers revisited the problem as new context-based information emerged [1],[12],[51]. The structure of a design has been shown to exist on a continuum, and a designer's knowledge of a problem imposed structure onto that problem [14].

### 2.4.5 Preferred Evaluation Criteria

The term "preferred evaluation criteria" refers to the minimal criteria a subject adopts to structure an ISP and guide the search for a satisfactory solution [13].

The development of Preferred Evaluation Criteria is a subset of problem restructuring [13]. The preferred evaluation criteria occurred on an individual level or group level [12],[51],[8].

Caution must be taken in selection of preferred evaluation criteria. Properly chosen criteria reduced design problem complexity, but poorly chosen criteria led to early reduction of the design problem and a closed minded approach to the situation [13],[12],[51].

### 2.4.6 Group Interactions

Group interactions is the dynamics of group work in software design. Motivation for the examination of group interactions arises from vast amounts of time designers spend in informal communication [15]. The terms "distributed" and "shared" cognition suggest that individual mental models coalesce via coordinated group action, resulting in a common model [12],[51]. The coalescing may result from the domination of a small coalition of individual designers (or sometimes just one designer) controlling the direction of the project [8].

### 2.4.7 Limitations of Design Studies

The primary limitation of the software design studies cited above is the sample size of each study. Most of these studies compared or examined less than ten participants performing software design; thus the external validity of these studies is weak.

## 2.5 Qualitative Studies

Much of software engineering experimentation is quantitative [18]. Qualitative research, however, affords subjects the opportunity to thoroughly and accurately detail their points of view concerning the topic at hand [36]. Qualitative inquiry also focuses on preserving the context of a situation, leveraging context as critical to understanding [36]. Qualitative results are derived from three types of data-collection: open-ended interviews, direct observation and written down document. All of these types typically come from fieldwork [36].

## 3. THE STUDY IN PROGRESS

We outline the qualitative empirical study. The objective of the study is to develop a descriptive model of how software designers make design decisions on different tiers. We begin with some definitions used in the hypotheses.

### 3.1 Definitions

We provide definitions for the following five terms: formal knowledge lookup, mental model change, optimal, participant and sub-goal.

#### 3.1.1.1 Formal Knowledge Lookup

Accessing some physical resource such as a text book, paper, personal notes, or the Internet to retrieve new information or confirm old information. Does not extend to learning from formal or informal discussions with colleagues or experts, which is covered in the interactions between designers.

#### 3.1.1.2 Mental Model Change

The change of a participant's mental model of a design as a result of an interaction with colleagues or experts or formal knowledge lookup.

#### 3.1.1.3 Potential Resolution

Agreement among the involved participant(s) as to what "should" be done. Does not extend to code optimization.

#### 3.1.1.4 Participant

Each of the software designers involved in the decision.

#### 3.1.1.5 Sub-Goal

Any desire secondary to the initial task at hand that influences the selection of a decision's alternative over another alternative.

### 3.2 Hypotheses

The hypotheses are presented in Table 5. They are divided by the type of design decision made, as previously defined, and by ten categories. The questions related to the categories are provided in Table 4.

### 3.3 Materials

The subjects of the study will be software developers, in the general sense of the term. Anyone who has participated in the design, coding and testing of software external to a course taken as part of an undergraduate degree program or the like is eligible to participate in the study. The sampling strategy is a combination of snowball sampling (asking subjects who else should be interviewed) and emergent sampling (opportunistic) [36].

### 3.4 Procedure

Three evaluation steps make up the study. Each of the three steps differs from the other two steps tiers in approach. The first step of evaluation is inductive interviewing, in which subjects are interviewed with no preconceptions concerning the direction of the research. The second step of evaluation is a mix of inductive and deductive interviewing, in which subjects interviewed and the interviewing technique are impacted by lessons learned from the first tier of interviewing. The proposed interviewing technique is the Critical Decision Method (CDM) [24]. The CDM begins with an initial open-ended question to provoke the subject to describe one or more design decisions. Following this description, the interviewer probes the subject with specific questions concerning one or more of the described incidents [24]. There are two units of analysis: the subject and the subject's decision making. The CDM guides control of the interview direction [24].

Table 4: Definitions of Hypotheses Categories

Options	How many options were generated?
Time	How long did it take to make the decision?
Interaction	Did the decision involve any interaction with a colleague?
Number of Interactions	If an interaction occurred, how many times did it occur?
Impact of Interactions	Did an interaction cause a mental model change in a participant? If so, in how many participants did the change occur?
Knowledge	Did formal knowledge lookup take place for a decision to be made?
Sub-Goal	Was a sub-goal considered or did a sub-goal direct the decision?
Potential Resolution	Was an agreed upon optimal identified?
Potential Unknown Optimal Searched	Did a search occur for an unknown optimal?
Identified Optimal Not Taken	Were the participants knowledgeable of an optimal that they then did not choose?

The final step of evaluation is inductive and deductive observation. At the stage we will observe design decisions as they occur in a software development organization. There will be three units of analysis: the subject observed, the group observed and the decision making. The interviewing technique only allows for a view of group dynamics through an individual perspective. Observation allows an examination of group dynamics in progress. By definition, there will be little control of the processes observed.

### 3.5 Limitations of the Study

Two initial limitations of this study have been identified. Our background and perspective has the potential to impact a qualitative study. We must be careful not to allow research biases to affect the direction of the interview, the observations recorded and the analysis attained. Secondly, snowball sampling is based on a "who-knows-who" concept. This leads towards a sample of the same types of developers, so that there is a greater likelihood that the participants will think alike and act alike. The backgrounds of the participants must be recognized.

### 3.6 Comparison & Benefits

This work differs from previous ethnomethodological studies that it proposes a much larger sample size than those previously presented. We are targeting 100 interviews and observations at 5-10 software development organizations.

## 4. CURRENT STATE

Twenty-seven interviews have been performed to date which make up the first step of evaluation: inductive interviewing. Content analysis is being used to analyze interview transcripts for common themes and frequency of themes [28]. An interactive dictionary is used to collect common words and phrases by continuously updating the dictionary based on new content taken from the interviews [28].

## 5. CONCLUSIONS

We presented the background work in software design, decision making, design rationale, software design studies and qualitative

studies. The definitions provided in Tables 1 & 2 will be used to group interview and observation results. Analysis of the inductive interviewing is currently underway using content analysis.

**Table 5 Hypotheses**

Design Decision	Implementation	Structural	Interaction	Architectural	Usability
Options	Will involve the consideration of no more than 1 option.	Will involve the consideration of at least 2 concurrent options.	Will involve the consideration of at least 2 concurrent options.	Will involve the consideration of at least 2 concurrent options.	Will involve the consideration of at least 2 concurrent options.
Time	Will be made in no more than 1 minute.	Will be made in no more than 10 minutes.	Will be made in no less than 1 hour.	Will be made in no less than 1 hour.	Will be made in no less than 1 hour.
Interaction	Will involve at most 1 other person.	Will involve at least 2 people.	Will involve at least 2 people.	Will involve at least 3 people.	Will involve at least 3 people.
Number of interactions	Will involve at least 1 interaction.	Will involve at least 1 interaction.	Will involve at least 2 interactions.	Will involve at least 2 interactions.	Will involve at least 2 interactions.
Impact of Interactions	Will involve no change from an interaction.	Will involve at least 1 mental model change as a result of the interaction(s).	Will involve at least 1 mental model change as a result of the interaction(s).	Will involve at least 2 mental model changes as a result of the interaction(s).	Will involve at least 2 mental model changes as a result of the interaction(s).
Knowledge	Will involve no formal knowledge lookup.	Will involve formal knowledge lookup.	Will involve formal knowledge lookup.	Will involve no formal knowledge lookup.	Will involve no formal knowledge lookup.
Sub-Goal	Will be impacted by at least one sub-goal.	Will be impacted by at least one sub-goal.	Will be impacted by at least one sub-goal.	Will always be impacted by a sub-goal.	Will not be impacted by a sub-goal.
Potential Resolution	25% of the time.	75% of the time.	75% of the time.	100% of the time.	100% of the time.
Potential Unknown Optimal Searched	50% of the time.	50% of the time.	50% of the time.	0% of the time.	0% of the time.
Identified optimal not taken.	25% of the time.	50% of the time.	50% of the time.	25% of the time.	25% of the time.

## 6. REFERENCES

- [1] Adelson B, Soloway E; "The Role of Domain Experience in Soft. Design"; IEEE Trans. Soft. Eng V11 No.11 Nov. 1985.
- [2] Ahmed S, Wallace K, Blessing L; "Understanding the Differences Between How Novice & Experienced Designers Approach Design Tasks"; Res. Eng. Design 14, 1-11 2003.
- [3] Beach L.R; "Image Theory: Personal & Organizational Decisions"; In: Klein, Orasanu, Calderwood, Zsombok, Decision Making in Action; NJ: Ablex Pub. Corp.; 1993.
- [4] Brehmer B, Joyce G.R.B; "Human Judgment: the SJT View"; Elsevier Science Ltd; 1988.
- [5] Bruegge B; Dutoit A.H; Object-Oriented Software Engineering 2<sup>nd</sup> Ed. Prentice Hall NJ; 2004.
- [6] Carroll J.M; Scenario-Based Design; John Wiley & Sons Inc. NY; 1995
- [7] Conklin J, Begeman M; "gIBIS: A Hypertext Tool for Exploratory Policy Discussion"; ACM Trans. Office Information Systems, V6 No.4 303-331 October; 1988.
- [8] Curtis B, Krasner H, isco N; "A Field Study of the Software Design Process for Large Systems", Comm. of the ACM, V.31 No.11, November; 1988.
- [9] Demarco T, Lister T; Peopleware: Productive Projects and Teams; 2<sup>nd</sup> Ed. Dorset House Pub. Co. New York, NY; 1999.
- [10] Fisher G, McCall R; "JANUS: Integrating Hypertext with a Knowledge-based design Environment"; Proc. 2<sup>nd</sup> ACM Conf. Hypertext & Hypermedia; Pitts., PA, 105-117; 1989

- [11] Flyvberg B; *Making Social Science Matter: Why Social Inquiry Fails and How it Can Succeed Again*; Cambridge University Press 2001.
- [12] Gasson S; "Framing Design: A Social Process View of Information System Development"; Proc. Int. Conf. Information Systems, Helsinki Finland, 224-236; 1998.
- [13] Guindon R; "Knowledge Exploited by Experts During Software System Design"; Int. J. on Man-Machine Studies 33, 279-304; 1990.
- [14] Guindon R; "Designing the Design Process: Exploiting Opportunistic Thoughts" HCI 5, 305-344; 1990.
- [15] Herbsleb J, Mockus A; "Formulation and Preliminary Test of an Empirical Theory of Coordination in Software Engineering"; Proc. 9th Eur. Soft. Eng. Conf./ACM SIGSOFT Symp. on the Found. of Soft. Eng; Sept 1-5, 2003.
- [16] Highsmith J; "e-Business Application Delivery: Managing Distributed Project Teams"; Cutter Consortium August 1999.
- [17] Highsmith J; *Agile Project Management*; Addison-Wesley, Boston MA; 2004.
- [18] Juristo N, Moreno A.M; *Basics of Software Engineering Experimentation*; Kluwer Academic Pub, Boston MA; 2001.
- [19] Kahneman D, Tversky A; "Prospect Theory: An Analysis of Decision under Risk"; Econ. V47 No.2 263-292; 1979.
- [20] Kaner S; Lind L, Toldi C, Fisk S, Berger D; *Facilitator's Guide to Participatory Decision Making*; S. Kaner BC 1996.
- [21] Kant E, Newell A; "Problem Solving Techniques for the Design of Algorithms"; Information Processing & Management; V.20 No.1 97-118; 1984.
- [22] Karsenty, L; "An Empirical Evaluation of Design Rational Documents"; Proc. of SIGCHI Conf. on Human Factors in Computing Systems; 150-156 Vancouver, BC; 1996.
- [23] Klein G; "Recognition Primed Decisions"; Advances in Man-Machine Systems Research 5 47-92; 1989.
- [24] Klein G, Calderwood R, MacGregor D; "Critical Decision Method for Eliciting Knowledge"; IEEE Trans. on Sys., Man and Cybernetics; V.19, No.3, May/June; 1989.
- [25] Klein M; "Capturing Design Rationale in Concurrent Engineering Teams", IEEE Comp. V26 No.1 93-47, Jan 1993.
- [26] Klein GA, Orasanu J, Calderwood R, Zsombok CE, *Decision Making in Action*. Norwood, NJ: Ablex Pub. Corp.; 1993.
- [27] Klein G; *Sources of Power*, MIT Press Camb. MA; 1998.
- [28] Krippendorff K; *Content Analysis: An Introduction to its Methodology*; V.5 Sage Pub. Beverley Hills Lond.; 1980.
- [29] Lee J, Li K-Y; "What's in Design Rationale?"; In: Moran, Carroll. *Design Rationale: Concepts, Techniques, and Use*; Lawrence Erlbaum Associates, Mahwah NJ; 1996.
- [30] Lipshitz R; "Decision Making as Argument-Driven Action"; In: Klein, Orasanu, Calderwood, Zsombok, *Decision Making in Action*; NJ: Ablex Pub. Corp.; 1993.
- [31] MacLean A, Young R.M, Bellotti V.M.E, Moran T.P; "Questions, Options, Criteria: Elements of Design Space Analysis"; In: Moran, Carroll, *Design Rationale*; Lawrence Erlbaum Associates, Mahwah NJ; 1996.
- [32] Malhotra A, Thomas J.C, Carroll J.M, Miller L.A; "Cognitive Processes in Design"; Int. J. Man-Machine Studies 12 119-140; 1980.
- [33] Montgomery H; "The Search for a Dominance Structure in Decision Making: Examining the Evidence"; In: Klein, Orasanu, Calderwood, Zsombok. *Decision Making in Action* Norwood, NJ: Ablex Pub. Corp; 1993.
- [34] Olson G.M, Olson J.S, Storosten M, Carter M, Herbsleb J, Rueter H; "The Structure of Activity During Design Meetings"; In: Moran T.P, Carroll J.M, editors. *Design Rationale*; Lawrence Erlbaum Associates, Mahwah NJ; 1996.
- [35] Orasanu J, Connolly T; "The Reinvention of Decision Making"; In: Klein, Orasanu, Calderwood, Zsombok, *Decision Making in Action*; NJ: Ablex Pub. Corp.; 1993.
- [36] Patton M.Q; *Qualitative Research & Evaluation Methods* 3rd Ed.; Sage Publications, California; 2002.
- [37] Pennington N, Hastie R; "A Theory of Explanation-Based Decision Making"; In: Klein, Orasanu, Calderwood, Zsombok, *Decision Making in Action*; NJ: Ablex Pub. Corp.; 1993.
- [38] Phillips LD; "A Theory of Requisite Decision Models"; Acta Psychologica 56 29-48; 1984.
- [39] Poppendieck, M, Poppendieck T; *Lean Software Development: An Agile Toolkit*; Addison Wesley, Upper Saddle River, NJ; 2003.
- [40] Rittel H; Webber M; "Dilemmas in a General Theory of Planning"; Policy Sciences 4, 155-169; 1973.
- [41] Rugaber S, Ornburn S, LeBlanc R; "Recognizing Design Decisions in Programs" IEEE Software; 1990.
- [42] Saaty T.L; *The Analytical Hierarchy Process: Planning, Priority Setting, Resource Allocation*; McGraw-Hill, NY 1980
- [43] Shum S.B; "Analyzing the Usability of a Design Rationale Notation"; In: Moran T.P, Carroll J.M, *Design Rationale*; Lawrence Erlbaum Assoc., Mahwah NJ; 1996.
- [44] Siddal J; *Analytical Decision-Making in Engineering Design*; Prentice Hall Inc., Englewood Cliffs NJ; 1972.
- [45] Simon H; "A Behavioural Model of Rational Choice"; Quarterly J. of Econ.; V69 Issue 1 99-118; 1955.
- [46] Simon H; "The Structure of Ill Structured Problems"; Artificial Intelligence 4, 181-201; 1973.
- [47] Sonnetag S; "Expertise in Professional Software Design: A Process Study"; J. App. Psych. V83 No5 703-715 Oct 1998.
- [48] Stirling W.C; *Satisficing Games and Decision Making*; Cambridge University Press, Cambridge, UK; 2003
- [49] Tversky A; "Elimination by Aspects: A Theory of Choice"; Psychological Review; V.79 No.4, 281-299; 1972.
- [50] Tversky A, Kahneman D; "Judgment Under Uncertainty – Heuristics and Biases"; In: Kahneman, Slovic, Tversky. *Judgment Under Uncertainty* Camb. Uni. Press; 1982.
- [51] Walz D.B, Elam J.J, Curis B; "Inside a Software Design Team: Knowledge Acquisition, Sharing and Integration"; Comm. of the ACM, V.36 No.10 Oct; 1993.