Documenting Design Decision Rationale to Improve Individual and Team Design Decision Making: An Experimental Evaluation

Davide Falessi
Univ. of Roma "Tor Vergata", DISP
Rome - Italy
+39 0672597942
falessi@ing.uniroma2.it

Giovanni Cantone
Univ. of Roma "Tor Vergata", DISP
Rome - Italy
+39 0672597392
cantone@uniroma2.it

Martin Becker
Fraunhofer IESE
Kaiserslautern - Germany
+49 63168002246
Martin.Becker@iese.fraunhofer.de

ABSTRACT
Individual and team decision-making have crucial influence on the level of success of every software project. Even though several studies were already conducted, which concerned design decision rationale documentation approaches, a few of them focused on performances and evaluated them in laboratory. This paper proposes a technique to document design decision rationale, and evaluates experimentally the impact such a technique has on effectiveness and efficiency of individual/team decision-making in presence of requirement changes. The study was conducted as a controlled experiment. Fifty post-graduate Master students performed in the role of experiment subjects. Documented design decisions regarding the Ambient Intelligence paradigm constituted the experiment objects. Main results of the experiment show that, for both individual and team-based decision-making, effectiveness significantly improves, while efficiency remains unaltered, when decision-makers are allowed to use, rather not use, the proposed design rationale documentation technique.

Categories and Subject Descriptors
D.2.m [**Software Engineering**]: Software Engineering Miscellaneous.

General Terms
Documentation, Design, Experimentation.

Keywords
Design decision rationale, Experimental evaluation, Individual and team decision-making.

1. INTRODUCTION
Individual and team decision-making have crucial influence on the level of success of any software project. Anyway, up to now, to our best knowledge, few empirical studies evaluated the utility of Design Decision Rationale Documentation (DDRD).

Several studies already have taken approaches and techniques to this end in consideration and have argued about their benefits, but only one of them [8] has focused on performance and has been evaluated it in laboratory.

ISESE'06, September 21-22, 2006, Rio de Janeiro, Brazil.

The contribution of this paper is twofold: (i) it briefs, for a larger audience, on the "Decision Goals and Alternatives" (DGA) DDRD technique, which was presented in form of a Technical Report [10] so far, and (ii) experimentally evaluates that technique with respect to the current practice of not documenting design rationale at all. The study was conducted as a controlled experiment with post-graduate Master students in the University of Rome "Tor Vergata".

The remainder of this paper is structured as follows: Section 2 presents motivation, view, goal and hypotheses of the conducted study. Section 3 considers related works. Section 4 introduces to DGA DDRD technique. Section 5 addresses experiment planning and operation issues. Sections 6 shows experiment results and data analysis, and Section 7 discusses results. Section 8 argues about threats to validity. We conclude this paper in Section 9 with some final remarks and prospective works.

2. STUDY MOTIVATION, VIEW, GOAL AND RESEARCH HYPOTHESES
The growing interest in software engineering decision support field [26] [27] reveals the crucial influence of decision-making on the level of success of any software project.

Because separately good decisions might be competitive, inconsistent or also in contradiction when viewed as a whole, software research still assigns great importance to individual and team decision-making, and agility in software process, as demonstrated by the relevant number of related studies [4, 5, 12, 18, 23, 24]. Concerning those issues, our conjecture is that DDRD is useful for individual and team decision-making in case of changes in requirements.

Formally, according to the GQM template [3], the goal of the presented study is *to analyze* the DGA DDRD technique *for the purpose of* evaluation *with respect to* effectiveness and efficiency of individual-decision-making and team-decision-making in case of changes in requirements *from the point of view of the* researcher *in the context of* post-graduate Master students of software engineering. From the motivations and goal above, the following research null hypotheses (resp. alternative hypotheses) follow for the presented study. After changes in requirements, when using or respectively not using DGA documentation of taken design decisions, there is no significant difference ($H_{0\_}$) (resp. significant difference, $H_{1\_}$) for individuals ($H_{\_I}$) (resp. teams, $H_{\_T}$), in the

amount of time needed for (hence efficiency, $H_{\_Ey}$) (resp., in the correctness, hence effectiveness of, $H_{\_Es}$) decision-making. Consequently, we have four null (resp. alternative) hypotheses, which we denote by H_{0IEy}, H_{0IEs}, H_{0TEy}, and H_{0TEs}, respectively (resp. H_{1IEy}, H_{1IEs}, H_{1TEy}, H_{1TEs}).

3. RELATED WORK

3.1 Design Decision Rationale

According to Lee [21], "Design rationales include not only the reasons behind a design decision but also the justification for it, the other alternatives considered, the tradeoffs evaluated and the argumentation that led to the decision".

Depending on the category of information documented, DDRD can be based on argumentation, history, device, process and active-document [7]. In the argumentation-based DDRD, design rationale is principally used to represent the arguments that characterize a design, like issues raised, alternative responses to these issues and arguments for and against each alternative. Prominent argument-based DDRD techniques are gIBIS [9], DRL [21] and QOC [22]

In our best understanding, while the only disadvantage of adopting a DDRD is the effort required to write and maintain the documentation, the advantages include: design verification, design evaluation, design maintenance, design reuse, design teaching, design communication, design assistance, and design documentation [7]. However, "despite the recognition of the importance of capturing and reusing architecture knowledge, there is no suitable support mechanism" [1].

Brathall et al. [8] presented a controlled experiment to evaluate the importance of DDRD when predicting change impact on software architecture evolution. Results show that DDRD clearly improves effectiveness and efficiency. However, such improvement was only partially statistically significant and authors explicitly called for further investigations. Between the present study and the one proposed by Brathall et al. there are similarities concerning dependent variables and methods for data analysis, and dissimilarities concerning many points, including requirement-change causes, types of subjects, type of decision to manage, and type of DDRD.

Karsenty [16] proposed an empirical evaluation concerning the utility of design rationale documents in maintenance of a nine months old software project. Between the present study and the one proposed by Karsenty, there are similarities concerning the purpose, and dissimilarities concerning many points, including requirement-change causes, types of subjects, objects, treatments, dependent variables, data collection mechanisms, and methods for data analysis.

Shum and Hammond [30] presented a good essay on pros and cons of adopting different argumentation-based DDRD.

3.2 Cooperation

Wu, Graham, and Smith [34] used interviews, shadowing, and communication-event-logging as data collection methods. Main results show that, concerning designers: 1) they communicate and collaborate by a wide variety of means; 2) they prefer general-purpose tools rather than domain-specific tools; 3) they change frequently their meeting place.

Seaman and Basili [29] studied the influence of organizational and process characteristics on the effort spent in collaboration. They used real-time observations and structured interviews. Their main result was that several organizational factors significantly affect communication effort (time spent).

Bellotti and Bly [4] emphasized on the importance both of local mobility in collaborative software design, and face-to-face style of communication.

Kraut and Streeter [18] utilized questionnaires and interviews for observing inter-team coordination practices. The major result was that developers cited discussion as their preferred communication means.

In the middle of 90's, Perry, Staudenmayer, and Votta [25] conducted two experiments revealing that a large percentage of the software process cycle was devoted to organizational concerns.

Tang [31] used videotapes to observe activities enacted by small teams in controlled environments. His major result was that the process of creating and using drawings conveys much information that is not captured within the drawing itself.

Concerning computer supported cooperative work, several studies were conducted focused on evaluating benefits on geographically distributed development [6]. Anyway, a very recent study [23] clearly reveals what other studies [4, 6, 8, 18, 23, 26, 27] state implicitly: measuring collaboration is hard.

3.3 Agility

Nowadays, agile software development has become extraordinarily fashionable. Agile methodologies include tools, processes, and approaches; they aim at helping software organizations in reacting to requirement changes, using the principles exposed in the Agile Manifesto. Major agile methodologies are: Extreme Programming, Scrum, Lean Development, Crystal and Context Driven Testing [12] [24]. However, the implementation of agile processes in traditional development organizations entails several management challenges [5].

4. THE DDRD DGA TECHNIQUE

Following the definition of design rationale provided by Lee [21] (see Section 3.3) the information that we want to document are the reasons why a decision has been taken. The SEI presents its Cost Benefit Analysis Method (CBAM) method [17] as a rational decision-making process for software architectural decisions, which is able to give stakeholders help in the elicitation of costs and benefits. The CBAM is based on attributes such as the importance of each objective for the project, the alternative decisions available and, for each alternative, to what extent that alternative fulfills those objectives. The basic principle is that each objective has its own level of importance and each alternative decision has, for each objective, its own level of fulfillment. The level of benefit related to an alternative is measured by summing the products of the level of fulfillments of each objective and the level of importance of that objective.

During the development of an Ambient Intelligence application, while trying to apply CBAM, the needs of improving collaboration among design decision makers arose, refining the grain of requirements traceability, and optimizing the usage of new technologies. Therefore, we eventually came to instantiate DDRD in

a specific documentation technique, the abovementioned DGA, which is driven by the decision goals and available design alternatives.

In the DGA technique, DDRD consists in documenting the attributes of CBAM. According to DGA, whatever the software context might be, design decisions depend on basic decision goals and inter-decision relationships, as shown in Table 1. In our experience, the set in Table 1 is complete, i.e. its elements are sufficient to specify the rationale of any software design decision. In the remaining, we call them Decision Goals.

Table 1. Entities influencing the rationale of design decisions ("Decision Goals")

Functional requirements
Non-functional requirements (quality attributes and constraints)
Business goals
Decision relationships

DGA not only aims to document already made decisions, but also to help decision-makers in making their further decisions. In DGA, the entity Decision is refined into two sub-concepts: Decision Type (DT) and Decision Alternative (DA). DT addresses the problem the decision should solve (e.g. What is the programming environment to use?) DA represents an available option (e.g. .NET). Two main insights drove the structure of the DGA technique: the importance of a goal is a DT attribute, while the goal fulfillment is a DA attribute. As a result of this clear separation of concerns, the maintainability of DDRD increased by avoiding document erosion and improving efficiency of document maintenance. For instance, a change in technology would affect DA only (level of fulfillment), while a change in requirements would affect DT only (level of importance). In fact, it is this separation of concerns that distinguishes DGA from other DDRD techniques.

According to DGA, in order to produce documented decisions, DDRD consists of two stages: (i) understand what to document, and (ii) enact the documentation.

The activities of the first stage consist in refining the project objectives and constraints, and comprehending which decision relationships are appropriate for the project. The refinement of the decision goals in Table 1 in sub-goals depends on the specific usage context. In fact, we provide DGA users with much more than Table 1: a framework for decomposing higher-level goals that prevents lacunae, and avoids misunderstandings.

The last stage is arranged in tasks: there is an instance of such a task for each design decision to make. Decisions makers can work in parallel (hence, tasks can be enacted in parallel). A task is arranged in three sequential blocks of activities, aimed to evaluate the "score" to give to relevant attributes of the current decision, for instance the priority of each objective in the designer view. The first block includes: (i) describing the current decision by providing information for the current DT, (ii) giving a score to each objective, to express the objective's importance for the current DT, and explaining motivations The second block includes: (i) describing each alternative of the current DT by providing more specific information, and then, (ii) for each objective, scoring to what extent the current alternative fulfills this objective, and (iii) for each relationship of the current alternative with alternatives of other DT(s), scoring to what extent the current alternative depends on each of the related DT(s), in the designer view. The last block selects in case the best alternative decision for the current DT and documents the alternative selected for the current decision. For further details about DGA we refer to [10].

4.1 Expected Effects on Collaboration

Improvement of collaboration among designers can be achieved by:

- Making team members aware of newly made design decision occurrences. In fact, this helps to detect, respectively avoid, conflicts between design decisions. The description of the expected relationships between decision alternatives and other decisions (see above for DGA activity concerned with scoring relationships between alternatives of different DT(s)) should provide helpful hints to this end.
- Allowing team members to share the characteristics of design decision alternatives. In fact, this exploits the different points of views of the various members, and consequently helps decision makers to consider the different view points. The quantification to what extent a decision alternative fulfills the objectives of a decision type (see above for DGA activities concerned with scoring fulfillment of objectives) should foster this.

Furthermore, DGA intends to improve the collaboration between designers and project manager by:

- Preventing designers' misjudgment of the goals' importance. Quantifying to what extent objectives are important for DT (see above for DGA activities concerned with prioritizing objectives) should allow managers to detect and resolve misinterpretations quickly.
- Identifying the violated requirements that cannot be met based on the already made design decisions. Quantifying to what extent decision alternatives fulfill the objectives (see again objective prioritization) should help to achieve this goal. Be aware that this additionally improves communication between product managers and customers, as the impact of a request is made explicit and thus foster the argumentation.

4.2 Expected Effects on Agility

Changes in the requirements necessitate changes in the objectives. Technological changes, in their turn, extend (usually) the set of available alternatives and help designers in recognizing better alternatives (if any). As DGA quantifies to what extent a decision alternative fulfills the objectives of a decision type (see DGA activity 2.4 above), it is expected to support agility issue. In fact, in case of changing requirements or technology, designers should be able to identify affected decisions (and artifacts) quickly and to reconsider them efficiently

5. EXPERIMENT PLANNING AND OPERATION

5.1 Experiment Definition and Setting

According to the study hypotheses and goal (see previous Section 2), we conducted a controlled experiment at the University of Rome "Tor Vergata", with fifty post-graduate local Master students performing in the role of experiment subjects. Design decisions regarding an AmI [32] prototype developed at Fraunhofer IESE constituted the experiment objects. The first author stayed six months in Kaiserslautern for studying the application domain and

reasoning about feasibility and other characteristics of the presented study. The gained experience helped the experimenters (i.e. the experiment research team members) to carefully replicate the original context in the experimental environment of this study, which improves the external validity of the study.

According to Section 3.1, the evaluation of collaboration issues implies several intrinsic threats, which affect construct validity and internal validity [23]. According to Zelkowitz and Wallace [35], developing the study in laboratory would mitigate the impact of construct and internal threats. Additionally, both the usage of quite real objects, and the experience that experimenters matured in field, should mitigate the presence of external threats, which might derive from using a laboratory for conducting the experiment [35]. Our consequent design decision was to conduct a controlled experiment in a synthetic environment.

The context of the current study is off-line (an academic environment) rather than in-line, based on students rather than professionals, using domain-specific and goal-specific quite real objects (as synthesized from real ones) rather than generic or toy-like objects.

In order to replicate the context of real world decision-making, the experimenters defined a synthetic software project, able to show and emphasize those aspects, which are in focus for the presented study. That project regards a hospital management system, and AmI issues (e.g. resource constraints, heterogeneous sensors, etc.) characterize it. Moreover, it is supposed that: the software system is at the start point of its second iteration (of some iterative development process, e.g. RUP [19]); in the mean time, system requirements did change, and designers, who had taken design decisions during the previous iteration, moved away and are no more available for giving explanation to current designers. Concerning the requirements change, experimenters applied change-causes, which usually affect software requirements in real projects, like: 1) Variations in the industrial strategic partnerships; 2) Changes in functional and non-functional requirements, resulting from the customer experience in using the previous version of the product; 3) Technology advances.

In order to replicate the real world context for software teams, the experimenters designed five different roles for the participants, and planned to compose teams by using those roles, one team-member per role, and five people per team. They designated each role to manage two different design decisions: the one with DGA-documentation and the other one without DDRD. They planned to characterize each experiment subject, based on the participant's personal experience and preferences, and assigned a subject to perform in one role. Issues regarding the mapping from subjects to roles are further described in Section 5.3.

The experimenters specified ten different design decisions to use as experiment objects and assigned each of them a unique integer number ranging from 1 to 10 as an ID. Subjects performing in the same roles were specified to fall in different teams and manage the same couple of decisions. Vice versa, all the subjects performing in different roles were assigned to have two different decisions to manage. Concerning the decisions assigned per subject, one was without design rationale documentation, i.e., only the result of the decision was reported (not its why): let us denote such a level with "non–DGA-documented". The second decision was documented with DGA, i.e. the documentation additionally included the decision rationale.

Due to the nature of the experiment object – the AmI prototype – and planned requirement changes, decision making concerned two types of components: the Central Computation Node (CCN) and the Personal Digital Assistant (PDA) component. Driven by this fact, the experimenters partitioned experiment design decisions in two classes: five decisions concerning CCN and PDA each. Of course, both classes were arranged in two versions: the one DGA-documented and the one non-DGA-documented. In particular, based on the experiment arrangement, decisions with ID(s) from 1 up to 5 fell in the first class and those with ID(s) from 6 up to 10 fell in the second class. Additionally, decisions belonging to the same class were interrelated with each other. For instance, the decision regarding the authentication mechanism to be used in a specific system component is strictly related with the decision concerning the physical capabilities of the component.

Based on the number of participants (50) and the team size (5), ten teams were organized randomly.

The experiment material was arranged on paper supports, and it regarded each decision to make by describing the requirements of both the previous (first) iteration and the current one (second iteration) of the project.

The experiment was balanced and the assignment of treatments to subjects was randomized. A decision was assigned to the same number of roles, subjects and teams. Moreover, each decision had the same number of instances (and treatments, DGA/non-DGA documented decisions). Each team received both classes of decisions, those concerning CCN and PDA, respectively. Five out of ten teams had DGA (resp. non-DGA) -documented decisions to manage that belonged to one class. The remaining five teams had DGA-documented decisions to manage that belonged to the other class.

Two main phases characterize the experiment. Both of them aimed at evaluating efficiency, effectiveness, and perceived utility of using/not using DGA DDRD in individual/team decision making in case of requirements change.

During the first phase of the experiment, each subject received two decisions, depending on the subject's role, and managed those decisions without cooperating with other participants. Those decisions belonged to different classes. Depending on the subject's team, one out of two decisions was DGA-documented. 50% of subjects had DGA-documented decisions to make first.

During the second phase, subjects convened with their teams and all together, discussed, for acceptance, each decision made during the first experiment phase. In fact, each decision was reconsidered. Each team member contributed to improve and enhance compatibility among decisions by valorizing at team level the subjective understandings about the available alternatives.

In each phase, the experiment material guided subjects to manage decisions assigned to them in a specified order. In the initial phase, each subject handled first the decision with the minimum ID and then the remaining decision. During the second phase, team members considered and finalized a half of their decisions (ID 1-5) first. Afterwards, they passed to evaluate the remaining decisions (ID 6-10). In our case, which included relationships between decisions, the experiment setting allowed us to evaluate decision

making per team by using five objects, each under two levels for the factor: DGA/non-DGA documented decisions.

Concerning the experiment validity threats, some considerations should be made at this point. In order to keep the impact of subjectivity in control, we designed a balanced experiment. Moreover, we applied both treatments to each subject. Furthermore, in order to keep in control the impact of learning effect on the experiment results, the treatments to use first were as much as to use last. Finally, because learning effects play a predominant role in decision making, we discarded the idea of using paired design for improving validity of the experiment data.

Table 2 shows the experiment setting, i.e., the structure of treatments, objects, subjects, and teams. Each row describes a team. An item in the first column shows the corresponding list of DGA-documented decisions (default decisions were non-DGA-documented). An item in the second column indicates the ID of the corresponding team. The rest of Table 2 is organized in five batches, each including tree columns, which represent the ID(s) of the experiment subjects, the ID(s) of decisions assigned to, and role played by each of those subjects, ordered from left to right each. For instance, the firsts row in Table 2 shows that subject with ID 43 played the role AS ("Software Architecture & Service Discovery"), addressed decisions 1 and 6, and belonged to team A, which included subjects with ID(s) 12, 50, 2, and 10.

Three further people participated in the experiment: one playing the role of the Application System General Manager, the other ones performing in the roles of Experiment Designer and Observer, respectively. The General Manager answered questions concerning the business strategy of the software organization. The Experiment Designer provided subjects with face-to-face technical explanation related to the usage of experimental objects. The Observer enforced subjects to respect rules, in order to have collected data acceptable for filtering and analysis.

5.2 Training

In order to enable the subjects to attend our DDRD experiment with enough confidence, we trained all of them through three plenary training sessions for a total of eleven hours. The first session of two hours was theoretic. The experimenters explained DDRD and AmI related issues. The second session took four hours. Here the experimenters performed in the role of decision makers and discussed five exemplary design decisions. The third session lasted five hours and was a trial of the experiment discussed in the presented work. For such a training session, all the characteristics were similar to those we would have utilized at experiment conduction time, less the application domain (a house AmI application was used for training) and related decisions. The experimenters checked that every subject was trained in every session sufficiently.

Table 2. Setting objects, treatments, subjects, and teams

Doc.	Team	Sub.	Dec.	Role	Sub.	Dec.	Role	Sub.	Dec.	Role	Sub.	Dec.	Role	Sub.	Dec.	Role
1,2,3,4,5	A	43	1, 6	AS	12	2, 7	CO	50	3, 8	DS	2	4, 9	HW	10	5, 10	IN
1,2,3,4,5	B	38	1, 6	AS	18	2, 7	CO	39	3, 8	DS	13	4, 9	HW	35	5, 10	IN
1,2,3,4,5	C	22	1, 6	AS	23	2, 7	CO	33	3, 8	DS	32	4, 9	HW	16	5, 10	IN
1,2,3,4,5	D	34	1, 6	AS	28	2, 7	CO	26	3, 8	DS	45	4, 9	HW	21	5, 10	IN
1,2,3,4,5	E	47	1, 6	AS	29	2, 7	CO	25	3, 8	DS	37	4, 9	HW	5	5, 10	IN
6,7,8,9,10	F	42	1, 6	AS	40	2, 7	CO	19	3, 8	DS	24	4, 9	HW	36	5, 10	IN
6,7,8,9,10	G	7	1, 6	AS	41	2, 7	CO	15	3, 8	DS	4	4, 9	HW	27	5, 10	IN
6,7,8,9,10	H	30	1, 6	AS	44	2, 7	CO	14	3, 8	DS	3	4, 9	HW	6	5, 10	IN
6,7,8,9,10	I	31	1, 6	AS	46	2, 7	CO	9	3, 8	DS	11	4, 9	HW	17	5, 10	IN
6,7,8,9,10	L	48	1, 6	AS	49	2, 7	CO	8	3, 8	DS	1	4, 9	HW	20	5, 10	IN

5.3 Subjects

Fifty attendees of the Experimental Software Engineering post-graduate course in their second and last year of Master Degree, participated in our work as experiment subjects, performing in the role of decision makers. While most of those subjects had already had some experiences at software companies, only few can be considered as software professionals. According to the classification scheme proposed by Höst et al. [13] experience and incentive of subjects can be classified respectively as "Graduate student with less than 3 months recent industrial experience" (E2) and "Artificial project"(I2).

In order to approximate the structure of a work-team in the AmI domain as much as possible, we modeled five different roles for decision making, one for each of the following areas: (i) HW – Hardware, (ii) CO – Communication, (iii) AS – Software Architecture & Services Discovery, (iv) IN – Inference, and (v) DS – Data Storage. For each of these roles specific knowledge and experience were requested. Systems were viewed from a certain perspective, and responsibilities were in place for certain types of decisions. Concerning our experiment, subjects expressed their preference for each role, according to their previous experience and level of confidence with the responsibilities of a role, well in advance of the last training session. Afterwards, subjects were mandatory assigned to those roles, which maximized the total of the expressed preferences. Hence, we split those fifty subjects into ten teams, each including five subjects, one per role.

5.4 Objects and Materials

Concerning controlled experiments conducted in synthetic environments, threats to external validity are strong enough to play the role of the Achilles' heel due to the usage of a context, which is quite different from real ones. For this reason we invested significant effort, during experiment planning and design, in carefully synthesizing the objects we already had encountered in former AmI software projects. Additionally, in order to keep experiment decisions as close as possible to real software design decisions, and provide subjects with real requirements as well as real decisions to make, we utilized the Amigo project documentation [1] as source of inspiration.

Figure 1 and Figure 2 show the forms that decision makers had to fill in during the first phase and the second phase of the experiment, respectively.

Subject ID:	First Phase Form			
	Initial Time	Decision Description	Final Time	Useful? (0≡NO / 1≡YES)

Figure 1. First experiment phase form.

Team ID:	Second Phase Form							
	Initial Time	DS	IN	CO	AS	HW	Final Time	Useful? (0≡NO / 1≡YES)

Figure 2. Second experiment phase form.

Due to space constraints, we refer to [11] for further description regarding objects used during the experiment and training sessions.

5.5 Factor and Parameters

The type of design documentation was the experiment factor. As already mentioned, we used two levels for this independent variable: "DGA-documented", and "non-DGA-documented", respectively. We controlled at a constant level the remaining independent variables, like experience of subjects, experiment materials, environment, and complexity of the experiment object.

5.6 Dependent Variables

As a general note to this section, we want to remark that we analyzed individual and team decision making by using both quantitative and qualitative data and quantitative analysis methods. Before we proceed with the performance evaluation of DGA-documentation in respect to individual and team decision-making and reaction of designers to requirements change, let us consider efficiency and effectiveness in some details.

IEEE defines efficiency as "the degree to which a system or component performs its designated functions with minimum consumption of resources" [15]. Our experiment subjects were allowed to use as much time as they required. Moreover, there were two decisions to make per subject. Hence, we assumed the inverse of the decision time as the punctual estimator to use for the efficiency of a decision. We measured efficiency quantitatively. Subjects used a predefined structured document to record, in real-time, the "Initial time" when they started to address a decision and the "Final time" when they completed the decision (see Figure 1 and Figure 2). During the first and second phase of the experiment, the Experiment Observer checked in an unobtrusive way the correctness of data recorded by subjects. The third training section had offered us the possibility to test and improve the data collection mechanism that we finally used in the experiment. The subsequent analysis of data revealed that subjects heavily round off data when they have to record directly the amount of time they needed to make a decision. Consequently, subjects had to track their used time with two time registrations for each made decision in the presented experiment. The used time for each decision, then was computed as the difference between those time stamps.

SEI defines Effectiveness as "the degree to which a system's features and capabilities meet the user's needs". Based on such a definition, we measured effectiveness by the amount of "correct" alternatives selected in decision making. This inevitably rises the question about the "correct" alternative for a decision: Is it the mode or the expected one? From a statistical point of view, the more an alternative is selected, the more it is correct. From an industrial point of view, the correct decision is the one that maximizes the utility. However, in our case, choosing the most useful alternative (industrial view) would depend on our subjectivity. In conclusion, we do not have clarified yet, which of the previous metrics is more appropriate for measuring the correctness of a decision in the context of the presented study. Fortunately, for our case, based on decisions that our experiment subjects made, we observed that, for every experimental object, the modal alternative coincides with the alternative that maximizes the utility, at least to our judgment. To evaluate effectiveness, we utilized qualitative and quantitative data analysis methods. In fact, for each made decision, experiment participants (individuals and teams, respectively) qualitatively described each their decision by filling in the field "Decision Description" of their form during the first phase (see Figure 1), and the second phase (see Figure 2), respectively. Since decision makers had been trained well, they described their decisions at a uniform abstraction level, which on one side avoided the risk of having vague or deeply detailed descriptions (i.e. incomprehensible and incomparable reports), on the other side helped experimenters to synthesize on, and assign the same ID to semantically equivalent decisions.

For each decision to make, the experiment decision makers (i.e. individuals or teams) provided Subjective quantitative measures (Yes or No) of the actual (resp. expected) Utility (SU) of the DGA-documentation given (resp. non-DGA-documentation given) to them. In other words, for each subject and decision, in case of a DGA-documented decision, SU is the answer to the question "Did DGA result significantly useful to you while making this decision?" Otherwise SU is the answer to the question "In your opinion, would DGA help you to make this decision significantly?" Decision-makers recorded their SU answers by filling in a predefined field of the given form (see the last column of both Figure 1 and Figure 2). Such an additional measure of the utility allowed us to triangulate results that the experiment had given for the main utility measures: effectiveness and efficiency. According to Seaman [28], that triangulation of empirical results helped us to observe the experiment outcome from several views, hence improved results validity.

As we utilized paper-based materials for collecting the experiment data, data entry was done manually by the experimenters and checked several times afterwards.

6. EXPERIMENT RESULTS AND DATA ANALYSIS

6.1 Data Set Reduction

We removed individual decision-making data of eight subjects and team decision-making data of one team from the data set to be analyzed, because those participants had round off all their data. Keeping that data would decrease the results validity.

6.2 Descriptive Statistics

Concerning individual decision-making in case of requirements changes, Table 3 shows the decision time (in minutes), the percentage of correct decisions made, and the perceived utility.

Concerning team decision-making in case of requirements changes, Table 4 shows decision time, percentage of correct made decisions, and perceived utility. Note that herein a team decision results from five singular decisions, one per team member, which all members approved.

Figure 3 plots individual decision-making results regarding efficiency in case of requirements changes. Let us note that, due to data reduction, the experiment data are not completely balanced (decisions were made not by the same number of subjects). As a consequence, for instance, means showed in Figure 3 might differ from the ones obtainable from Table 3. Figure 4 shows team decision-making data results regarding efficiency. Figure 5 shows individual decision-making data results regarding effectiveness in case of requirements changes. Figure 6 shows team decision-making data results regarding effectiveness. Figure 7 shows individual data-results regarding the

perceived utility of decisions made in case of requirements changes.

As all teams perceived DGA-documentation for each of the decisions assigned to them as useful, we do not show plots concerning perceived utility for team decision-making.

Table 3. Average data for individual decision-making.

	Average	With Doc.	Without Doc.
Time Required (min.)	10	11	10
Correct Decisions (%)	75	86	64
Perceived Utility (%)	77	81	74

Table 4. Average data regarding collaboration.

	Average	With Doc.	Without Doc.
Time Required (min.)	18	17	19
Correct Decisions (%)	39	67	11
Perceived Utility (%)	100	100	100

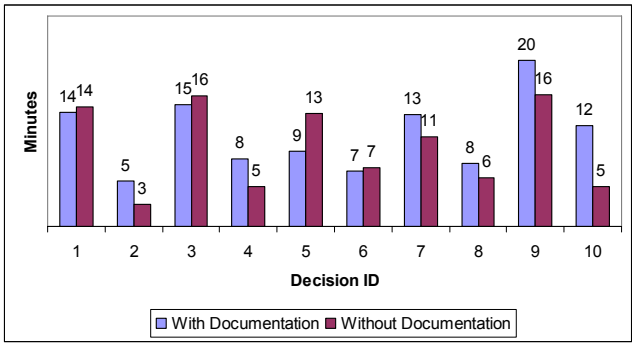

Figure 3. Efficiency in individual decision-making.

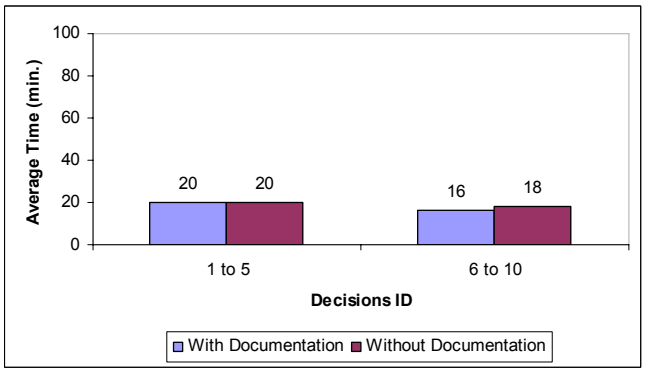

Figure 4. Efficiency in collaboration.

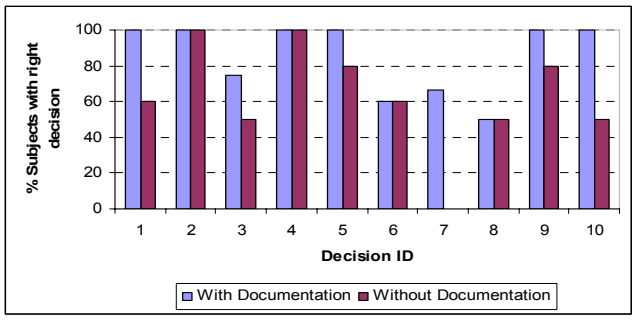

Figure 5. Effectiveness in individual decision-making.

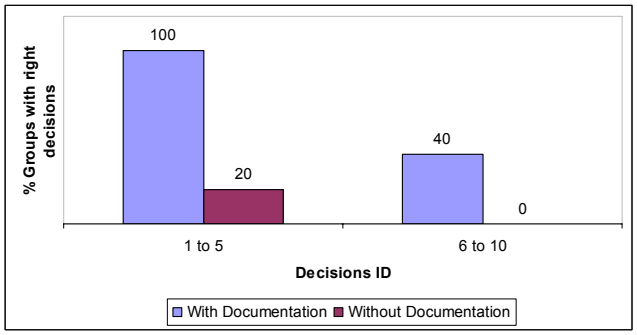

Figure 6. Effectiveness in collaboration.

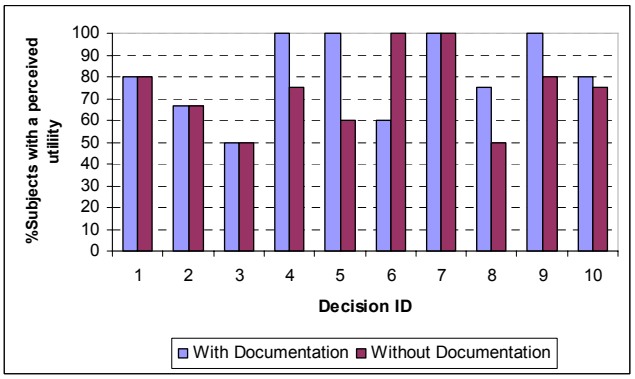

Figure 7. Perceived utility in individual decision-making.

6.3 Hypotheses Testing

6.3.1 H_{0IEy}: DGA documentation does not affect efficiency in individual decision-making

In order to test hypothesis H_{0IEy}, we compare two samples for decision-making after requirements change. Those samples concern decision time for formerly DGA-documented decisions and non-DGA-documented decisions, respectively. For the normality tests, which we applied to both the given data sets, the Shapiro-Wilks test provided, as P-values, 0.05532 for decision times of DGA-documented decisions, and 0.00003 for decision times of non-DGA-documented decisions. Because the latter is less than 0.01, we can reject the idea that such a distribution comes from a normal distribution with the 99% confidence level. Subsequently, for those samples of data, the Mann-Whitney test provides 0.22787 as P-value. Because such P-value is greater than 0.05, we can assert that *there is not* a statistically significant difference between the medians at the 95.0% confidence level. Hence, we cannot reject the null hypothesis H_{0IEy}.

6.3.2 H_{0IEs}: DGA documentation does not affect effectiveness in individual decision-making

In order to test hypothesis H_{0IEs}, we compare two samples for correctness of decisions made after requirements change by using DGA-documented decisions and non-DGA-documented decisions, respectively. Again, the Shapiro-Wilks test provided the lowest P-values (0.0000 in both cases) for the normality of those data sets. Since those P-values are both less than 0.01, we

can reject the idea, that any of those data samples comes from a normal distribution with the 99% confidence level. Moreover, for those data samples, the Mann-Whitney test provides 0.024558 as P-value. Because such P-value is less than 0.05, we can assert, that *there is* a statistically significant difference between the medians at the 95.0% confidence level. Hence, we can reject the null hypothesis H_{0IEs}.

6.3.3 H_{0TEy}: DGA documentation does not affect efficiency in team decision-making

In order to test hypothesis H_{0TEy}, we compare two samples of decision time for team decision-making, when using formerly DGA-documented and non-DGA-documented decisions, respectively. Again, the Shapiro-Wilks test provided the lowest P-values (0.811 and 0.109, respectively) for the normality of those data samples. Because both of these P-values are greater than 0.10, we cannot reject the idea that both distributions come from normal distributions with 90% or higher confidence. The P-value given by the F-Test is 0.849. Since such P-value is greater than 0.05, we can assert, that there is not a statistically significant difference between the samples standard deviations at the 95.0% confidence level. The P-value provided by the T-Test is 0.442307. Because such P-value is greater than 0.05, we can assert, that *there is not* a statistically significant difference between the means of the two samples at the 95.0% confidence level. Hence, we cannot reject the null hypothesis H_{0TEy}.

6.3.4 H_{0TEs}: DGA documentation does not affect effectiveness in team decision-making

In order to test hypothesis H_{0TEs}, we compare two samples regarding team decisions correctness, when using formerly DGA-documented and non-DGA-documented decisions, respectively. For both those data sets, the Shapiro-Wilks test provided the lowest P-values, 0.00024 and 0.00000, for DGA-documented and non-DGA-documented decisions, respectively. Because both of these P-values are less than 0.01, we can reject the idea, that both distributions come from a normal distribution with the 99% confidence level. The Mann-Whitney test provided 0.021610 as the P-value for those data sets. Such P-value is less than 0.05, which asserts, that *there is* a statistically significant difference between the medians at the 95.0% confidence level. Hence, we can reject the null hypothesis H_{0TEs}.

7. RESULTS DISCUSSION

7.1 Individual decision-making

Let us discuss now the impact of DGA documentation on the individual decision-making when requirements do change.

7.1.1 Efficiency

Based on results presented by Table 3, Figure 3, and Section 6.3.1, we can argue that the usage of DGA-documentation for design decisions *does not* significantly affect the decision time, when requirements do change.

7.1.2 Effectiveness

Based on results presented by Table 3, Figure 5, and Section 6.3.2, we can argue that the usage of DGA documentation significantly improves the effectiveness of decisions made when requirements do change. Figure 5 shows that effectiveness *improves*, when DGA documentation is utilized, whatever might be the type of decision to make. We highlight the fact, that no subject was able to make the right decision with ID 7 without DGA-documentation.

7.1.3 Subjective Utility

Based on results presented in Table 3, we can assert, that subjects felt *comfortable* with DGA documentation. In fact, they found it useful for 77% of decisions they made. Note that this result does not depend on the level of documentation utilized. However, for both levels of documentations, more than 50% of participants felt DGA as useful, as Figure 7 shows.

7.2 Team decision-making

In these subsections we discuss data regarding decision made by team of five people, when decisions were DGA-documented and non-DGA-documented, respectively.

7.2.1 Efficiency

From Table 4, Figure 4, and Section 6.3.3 we can argue, that the use of documentation *does not* affect the time needed in team decision-making.

7.2.2 Effectiveness

From Table 4, Figure 6, and Section 6.3.4 we reveal, that the use of DGA-documentation *affects* the effectiveness of decision-making in a team of five people in a significant and *positive way*. Figure 6 shows an important result: for every decision, the relative effectiveness is higher if DGA is applied.

7.2.3 Subjective Utility

Every team feels DGA *useful* in every fivefold decision.

8. Threats to Validity

In order to help the readers qualifying the results of the presented study, we discuss the way in which we mitigated validity threats [33].

8.1 Conclusion validity

Reliability of measure is achieved by a careful selection of the data collection mechanisms, metrics, and a checked data entry. Reliability of treatments implementations are achieved by balances and randomization implemented in the experiment (see Section 5.1).We can argue that, in our case, the heterogeneity among subjects did not affect data validity for two main reasons:

1) Subjects were divided in roles based on their best attitude. Based on their background and behaviors, students seemed to have the same level of specialization.

2) Each subjects applied both treatments. We avoided fishing activity by: i) defining experiment data analysis details (analysis method, level of significance, etc.) according to standards [33] and before running the experiment; moreover ii) describing results without any omission (see Section 6).

8.2 Construct validity

We mitigated mono method bias threats by using qualitative, quantitative, objective, and subjective measures. Our work seems to be not exposed to restricted-generalizability-across-constructs threats. In fact, we cannot find disadvantages in adopting DDRD less than time documenting decisions. Based on results from our third experiment-training session: It is seventeen minutes the average development time for decision documentation (from the scratch) by using DGA. After changes in requirements, it is tree minutes the average maintenance time for decision documentation by using DGA. Evaluation-apprehension and Hawthorne-effect did not threaten our experiment, because decision makers are subject to high pressure in the real world. We mitigated hypothesis-guessing threats by not spurring subjects on any treatment.

8.3 Internal validity

History threats did not show in the presented experiment, because treatments were applied one time per subject. Concerning maturation threats, subjects apparently stayed focused on their tasks. Moreover, as already mentioned, order of treatments is balanced. We mitigated instrumentation threats by refining collection forms, following suggestions that subjects gave to experimenters at training time. Characteristics and motivations of participants were enough for mitigating selection threats. In fact, subjects were at least as motivated as decision makers of real world software projects. Mortality threats did not affect the presented experiment, since no subject withdrew.

8.4 External validity

As we based our experiment on a synthetic environment, we cannot assure that the experiment objects represent the real world of software projects. Interaction of setting and treatment can be considered an important and unresolved (probably irresolvable) threat of this type of experiments. Nevertheless, it is important to highlight that in order to mitigate such threats the experimenters: 1) used all their experience with software projects in the real world, 2) reused data from documentation related to real software projects [1], 3) spent enough effort in designing and developing experiment objects. Additionally, we cannot assure that our experiment subjects are representative of, and can be compared for level of competence with, decision makers of real world software projects. It is important to highlight that the major part of subjects already had some experience in real world software industry. Moreover, experimenters gave roles to people in the purpose of maximizing the subjects' preferences and skill.

9. CONCLUSIONS AND FUTURE WORKS

This paper presented an experimental study aimed to evaluating the effects of Design Decision Rationale Documentation (DDRD) on individual and team decision-making in case of requirements changes. The Decision Goal Alternatives technique (DGA) was defined and used as DDRD instance.

Motivations for the presented study were: 1) Individual and team decision-making are two important issues in the development of software systems. 2) It is rational to expect that DDRD improves effectiveness and efficiency of both individual and team decision-making. 3) A previous study [8], which also investigated the improvement, differs from the present study in many aspects (e.g. type of decision, type of subjects and type of changes); moreover it explicitly called for further investigations.

In order to gain in validity of the experiment results, the experience of the experimenters allowed them to replicate carefully real-world AmI software projects in the adopted experimental synthetic environment. In order to facilitate the experiment replication we published materials and data concerning training sessions and experiment (see [11]). However, those few, potential, welcome people, who would replicate this work are advised that the experiment planning requested a quite huge effort.

The experiment main results derive from objective data and show that, in presence of changes in requirements, individual and team decision-making perform as in the following: (1) Whatever the kind of design decision might be, the effectiveness improves when DGA-documentation is available. (2) DGA-documentation seems not to affect efficiency. Regarding the utility of DGA, supplementary results, which are based on subjective data, allowed us to confirm the main results by a triangulation activity.

Concerning future works, our plan is to investigate how DDRD can be customized, depending on the usage context (e.g. business goal, domain, design method).

10. REFERENCES

[1] M. Ali Babar, I. Gorton,, R. Jeffery, "Capturing and Using Software Architecture Knowledge for Architecture-Based Software Development," Fifth International Conference on Quality Software (QSIC'05), 2005.

[2] Amigo project documentation, http://www.hitech-projects.com/euprojects/amigo/deliverables.htm

[3] V. Basili, G. Caldiera G. and D. Rombach, "Goal question metric paradigm" in Encyclopedia of Software Engineering, vol. 1, J. J. Marciniak, Ed.:John Wiley & Sons, 1994.

[4] V. Bellotti and S. Bly, "Walking away from the desktop computer: Distributed collaboration and mobility in a product design team", Computer Supported Cooperative Work, Cambridge, MA, ACM Press, 1996.

[5] B. Boehm and R. Turner, "Management Challenges to Implementing Agile Processes in Traditional Development Organizations", IEEE Software, Volume 22, Issue 5 (September 2005).

[6] U. Borghoff and J. Schlichter, "Computer-Supported Cooperative Work: Introduction to Distributed Applications", 2000, Springer.

[7] J. Burge and D. Brown, "Design Rationale Types and Tools", http://web.cs.wpi.edu/Research/aidg/DR-Rpt98.html,1998. Last access: 12/05/2006.

[8] L. Bratthall L., E. Johansson, B. Regnell, "Is a Design Rationale Vital when Predicting Change Impact? A Controlled Experiment on Software Architecture Evolution", Second International Conference on Product Focused Software Process Improvement, 2000.

[9] J. Conklin and M. Begeman, "gIBIS: a hypertext tool for exploratory policy discussion", ACM Transactions on Information Systems, v.6 n.4, p.303-331, Oct. 1988

[10] D. Falessi and M. Becker, “Documenting Design Decisions: A Framework and its Analysis in the Ambient Intelligence Domain”, BelAmI-Report 005.06/E, Fraunhofer IESE, March, 2006.

[11] D. Falessi, G. Cantone and M. Becker, “Materials and data concerning training sessions and experiment regarding DDRD”, http://ese.uniroma2.it/DDRD_Experiments.zip

[12] J. Highsmith, A. Cockburn, “Agile Software Development: The Business of Innovation”, Computer, v.34 n.9, p.120-122, September 2001.

[13] M. Höst, C. Wohlin, T. Thelin, “Experimental context classification: incentives and experience of subjects”, 27th international conference on Software engineering, 2005.

[14] M. Fowler, “The New Methodology”, http://www.martinfowler.com/articles/newMethodology.html, 2003.

[15] Institute of Electrical and Electronics Engineers, “IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries”. New York, NY: 1990.

[16] L. Karsenty, "An empirical evaluation of design rationale documents", Human factors in computing systems: common ground 1996.

[17] R. Kazman, J. Asundi and M. Klein, “Quantifying the Costs and Benefits of Architectural Decisions”, 23rd International Conference on Software Engineering, 2001.

[18] R. Kraut and L- Streeter "Coordination in Software Development", Communications of the ACM 38(3), 69-81.

[19] P. Kruchten, “The Rational Unified Process: An Introduction”, 3e. Addison-Wesley-Longman, 2003.

[20] J. Lee and K. Lai, “What's in Design Rationale?”, Human-Computer Interaction, 6(3&4), 251-280,1991.

[21] J. Lee, “Design Rationale Systems: Understanding the Issues”, IEEE Expert, Vol. 12, No. 3, pp. 78-85, 1997.

[22] A. MacLean, R. Young, V. Bellotti and T. Moran, “Questions, options and criteria: Elements of design space analysis”, Human-Computer Interaction, 201-250, 1991.

[23] D. Neale, J. Carroll and M. Rosson, “Evaluating Computer-Supported Cooperative Work: Models and Frameworks”. CSCW, 2004.

[24] R. Martin, “Agile Software Development, Principles, Patterns and Practices”, Prentice Hall, 2002

[25] D. Perry, N. Staudenmayer and L. Votta, “People, Organizations, and Process Improvement”, IEEE Software, July 1994, pp36-45.

[26] G. Ruhe, “Software Engineering Decision Support”, Special Issue International Journal of Software Engineering and Knowledge Engineering, Vol. 13, No. 5, Oct 2003.

[27] G. Ruhe, “Software Engineering Decision Support ? A New Paradigm for Learning Software Organizations”. Learning Software Organizations 2002: 104-113

[28] C. Seaman, “Qualitative Methods in Empirical Studies of Software Engineering” IEEE Transaction on Software Engineering 25(4): 557-572 (1999).

[29] C. Seaman and V. Basili, "Communication and Organization in Software Development: An Empirical Study." IBM Systems Journal 36(4), 1997

[30] S. Shum, S. Buckingham and N. Hammond, "Argumentation-Based Design Rationale: What Use at What Cost?" International Journal of Human-Computer Studies 40, 4 (April 1994): 603-52.

[31] J. Tang, "Findings from Observational Studies of Collaborative Work", International Journal of Man-Machine Studies, 1991.

[32] W. Weber, J. Rabaey and E. Aarts, “Ambient Intelligence”, Springer; 2005.

[33] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén: “Experimentation in Software Engineering: An Introduction”, The Kluwer International Series in Software Engineering, 2000.

[34] J. Wu, N. Graham and P. Smith, “A Study of Collaboration in Software Design”. International Symposium on Empirical Software Engineering 2003.

[35] M. Zelkowitz and D. Wallace, “Experimental Models for Validating Technology,” Theory and Practice of Object Systems, IEEE Computer, 31, 5, 23-31, 1998.