On the Effectiveness of Screen Mockups in Requirements Engineering: Results from an Internal Replication

Filippo Ricca
DISI, Università di Genova
Genova, Italy
filippo.ricca@disi.unige.it

Giuseppe Scanniello
DMI, Università della Basilicata
Potenza, Italy
giuseppe.scanniello@unibas.it

Marco Torchiano
Politecnico di Torino
Torino, Italy
marco.torchiano@polito.it

Gianna Reggio
DISI, Università di Genova
Genova, Italy
gianna.reggio@disi.unige.it

Egidio Astesiano
DISI, Università di Genova
Genova, Italy
astes@disi.unige.it

ABSTRACT

In this paper, we present and discuss the results of an internal replication of a controlled experiment for assessing the effectiveness of including screen mockups when adopting Use Cases. The results of the original experiment indicate a clear improvement in terms of understandability of functional requirements when screen mockups are present with no significant impact on effort. The data analysis of the replication, conducted also in this case with undergraduate students, confirms the results of the original experiment with slight differences, thus confirming that screen mockups facilitate the understanding of requirements without influencing the effort. We also sketch here some issues related to the documentation and communication between experimenters.

Categories and Subject Descriptors: D.2.1 [Requirements/Specifications]: Methodologies, Tools

General Terms: Experimentation, Measurement

Keywords: Empirical Studies, Internal Replication, Use Cases, Screen Mockups

1. INTRODUCTION

In the recent years, the software engineering community has defined a number of modeling methods/techniques to specify software requirements [13]. Among them, Use Cases are recognized as a simple way to capture and define requirements from the end user point of view. They are textually specified according to a more or less rigorous template, which generally enables the definition of a sequence of steps to describe the interaction between one or more actors and the system to develop. Some of the proposed methods/techniques introduce screen mockups to increase the comprehension of functional requirements and then promote the communication among stakeholders, analysts, and developers. However, for professional software engineers to consider adopting one method rather than another, they must know their effectiveness and weakness. An effective way to do it is resorting to empirical studies.

Cheng and Atlee in [8] point out that most empirical investigations in Requirement Engineering (RE) take the form of proofs-of concept or pilot studies. Although empirical studies have recently been conducted to assess the effectiveness of modeling techniques and to validate elicited requirements and software models [3, 19], the authors identified empirical studies as a "hot" research topic that should be better addressed in a near future.

Replications are however needed to increase the body of knowledge about the modeling techniques to adopt. Indeed, they are considered to be a crucial aspect of the scientific method since relevant and credible results can only be obtained by performing replications [6]. A single study rarely provides definitive answers. Unfortunately, replications are not so usual in the software engineering as shown in the study by Sjøberg *et al.* [24].

In this paper, we present the results of an internal replication of a controlled experiment for assessing the effectiveness of including screen mockups when adopting Use Cases. The subjects of the original experiment [21] were second year undergraduate students in Computer Science at the University of Basilicata (Italy). Post-questionnaires analysis of that experiment showed that the subjects judged useful the inclusion of screen mockups when adopting Use Cases. That result was also confirmed by the data analysis, which indicates a clear improvement in terms of understandability of functional requirements when screen mockups are present with no significant impact on effort.

To verify these results, we carried out a close replication [15] of that experiment involving a group of third year undergraduate students in Computer Science at the University of Genova (Italy). The data analysis of the replication has confirmed the results of the original experiment [21], thus increasing our awareness on the benefit deriving from the use of screen mockups. Slight differences, possibly due to the

ESEM 2010, September 16-17, 2010, Bolzano-Bozen, Italy. Copyright 2010 ACM 978-1-4503-0039-01/10/09...$10.00.

changes in the experimental context and to the time constraints in the execution phase, were observed in the results of the replicated experiment.

The paper is organized as follows. Section 2 discusses a subset of the related literature concerning experiments aimed at assessing the use of graphical elements in comprehension tasks. Section 3 provides an overview of the original experiment and then describes the design and execution of the replication. Section 4 presents the data analysis, while Section 5 discusses the results of the replication and the threats to validity. Final remarks and suggestions for future work conclude the paper.

2. RELATED WORK

In this section, we focus only on related work concerning controlled experiments aimed at assessing the comprehension of notations with and without graphical elements. A number of experiments have been performed to assess how graphical elements support the understanding of software artifacts at analysis, design, and code level. Almost all of them (except, e.g., [20]) show that the subjects benefit from the use of graphical elements.

A controlled experiment is reported in [22] to assess the impact of *Fit tables* on the clarity of requirements. Fit tables are simple HTML tables serving as the input and expected output for the acceptance tests. In that experiment, the students could only use the Fit tables to better grasp the requirements without the possibility of executing them. The results indicate that Fit tables help the requirements understanding, but need additional effort (even if not in significant way).

Gemino *et al.* [4] empirically investigate whether Use Case diagrams improve the effectiveness of Use Cases by providing visual clues. The investigation is conducted providing a group of students only Use Cases (control group) or Use Cases augmented with Use Case diagrams (treatment group). Results show that students employing both a Use Case diagram and Use Cases achieve a significantly higher level of understanding.

Staron *et al.* in [25] present a series of controlled experiments, performed with students and professionals, to assess the effectiveness of stereotypes in UML class diagrams to comprehend Object-Oriented applications in the telecommunication domain. They show that the use of stereotypes significantly helps both students and professionals to improve comprehension.

Ricca *et al.* [20] conduct an experiment with bachelor, master, and PhD students on the understanding of the UML Conallen's stereotypes in the context of the design of Web applications. Conversely, with respect to [25], stereotypes seem little useful in understanding. The main finding of that experiment is that stereotypes reduce the gap between experienced and less experienced subjects.

The usefulness of graphical elements to support understanding and maintenance tasks is also experimentally assessed by Bratthall and Wohlin [7], who compared ten different representations aiming at enriching the design of a software architecture with graphical elements. The subjects are master and PhD students.

Hendrix *et al.* [12] present two experiments to investigate the influence of additional graphical information on the source code. The focus of the work is to understand the effects of the Control Structure Diagram (CSD) on program comprehensibility. CSD is a graphical notation able to represent some constructs of programming languages (i.e., sequence, selection, and iteration) by means of graphical symbols. The statistical analysis of those data revealed that this notation improves the performance in program comprehension.

3. THE EXPERIMENTATION

In this section we first present an overview of the original experiment followed by the detailed design and execution of the internal replication. We adopted the guidelines proposed by Wohlin *et al.* [28] and Juristo and Moreno [14]. For replication purposes, the experimental package (in Italian) and the raw data of both the original and replicated experiment are available for downloading on the web[1]. A technical report [21] of the original experiment is also available there.

3.1 Overview of the Original Experiment

In the following we provide an overview of the context, design, and the results of the original experiment.

3.1.1 Experiment Context, Goal, and Design

The experiment was conducted within a research laboratory at the University of Basilicata with 33 second year undergraduate students in Computer Science. This experiment was an optional educational activity of a Software Engineering course.

The goal of the experiment consisted in assessing whether screen mockups provide a more effective way to increase the comprehension of functional requirements with respect to the only adoption of Use Cases. To this end, the subjects were provided with the requirements analysis documents of two different software systems: EasyCoin (a software for coin collection) and AMICO (a software for condominium management). The systems are similar in complexity and refer to application domains in which the subjects were not completely familiar with. The documents have been chosen as they are small enough to fit the time constraints of the experiment and realistic for small size projects (requirements documents count 19 Use Cases for AMICO and 20 for EasyCoin). According to the experiment design, steps within the scenario of a Use Case may be accompanied by screen mockups presenting what an actor will see in that moment. See Figure 1 for an example of screen mockup linked with the Use Case of Figure 2 (steps of the Use Case are numbered).

The design used was the *counterbalanced experimental design* [28]. In order to limit the learning effects among the experimental tasks, an exercise not directly related with these tasks was conducted some days before the experiment. The exercise was based on a requirement analysis document of a simple system named *LaTazza*. The screen mockups effects on the software requirements comprehension was evaluated

[1] www.scienzemfn.unisa.it/scanniello/ScreenMockupExp/

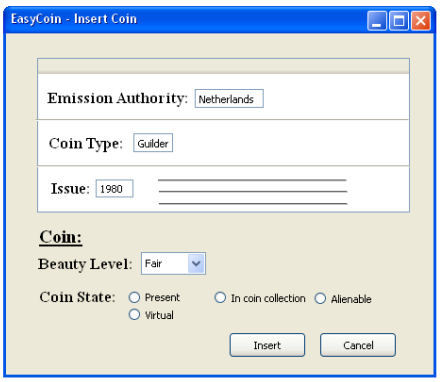

Figure 1: Example of screen mockup used in the experiment (EasyCoin system)

using a questionnaire based approach (as in [1, 11, 17, 20]). Instead, effort was measured in minutes.

USE CASE: Insert Coin
Level: User-Goal
Intention in context: the collector wants to insert a coin in the collection
Primary actor: coin collector
Precondition: a non-void list of issues is selected
Main success scenario:
1 the collector chooses an issue of the list and asks for inserting a coin
2 the system asks for coin info **4 (**see Insert Coin screen mockup**)
3 the collector inserts the info and presses insert button
4 the system shows the new inserted coin to the collector and the Use Case ends with success

Figure 2: Example of Use Case. (4) refer to the item 4 of the glossary (not shown here)**

3.1.2 Experiment Results

The results of the experiment provide strong evidence that the presence of screen mockups in Use Case descriptions significantly yields a large advantage in terms of comprehension level. The data analysis indicates that large benefits (46.5% improvement in comprehension level) are obtained adopting screen mockups. A further analysis revealed that the improvement for AMICO is slightly larger than for EasyCoin, when using screen mockups. While the problem domain of EasyCoin is quite popular, the subjects were less familiar with the domain of AMICO. This suggests that the degree of benefits attainable from the presence of screen mockups is inversely proportional to the *familiarity* of the subjects with the problem domain.

The average effort required in presence of mockups is slightly lower. We can generally state that the improvements in terms of comprehension level come at no expense in terms of comprehension effort. Also, for the effort we investigated the effect of subjects' familiarity on the problem domain of the software systems used in the experiment. A significant difference was only found for EasyCoin, while no difference was revealed for AMICO. In this case the *familiarity* with the application domain seems affecting positively the benefits achieved by adding mockups (i.e., if students are familiar with the application domain, the comprehension effort significantly decrease when mockups are used).

3.2 The Internal Replication

We present here a close [15] replication to confirm or contradict the results of the original experiment and to further investigate the influence of the problem domain of the software systems on the subjects' performances.

3.2.1 Context

Our replication was conducted within a laboratory at the University of Genova (Italy) with 51 third-year students of the Bachelor program in Computer Science. It represented an optional educational activity of a Software Engineering course. Previously, the subjects had attended and passed exams on basic and advanced object oriented programming and on database systems modeling.

The same requirements analysis documents used in the original experimentation have been considered also for the replication: EasyCoin and AMICO. These documents contained: *(i)* the mission of the software system; *(ii)* a UML Use Case diagram; *(iii)* the Use Cases (with or without mockups) specified following the SWEED template[2] (see Figure 2 for an example of SWEED template); *(iv)* a glossary including the needed definitions to comprehend the requirements specification.

It is important to point out that, the screen mockups (when present) provide no additional information that could not be derived from the requirements analysis document.

3.2.2 Hypotheses Formulation

Also in this replication we are interested in the Use Cases modeling as a communication tool among Analysts and Designers. In fact, a good comprehension of the functional requirements is vital for building the right system in the right time. Accordingly, the *perspective* of this study is from the point of view of the *Researchers*, investigating on the effectiveness of screen mockups when adopting Use Cases, and of the *Project managers*, evaluating the possibility of adopting Use Cases augmented with screen mockups in their organization. We have then investigated the following one-tailed null hypotheses:

H_{l0} When performing a comprehension task, the presence of screen mockups in Use Cases **does not significantly improve** the comprehension level of software requirements.

H_{e0} When performing a comprehension task, the presence of screen mockups in Use Cases **does not significantly reduce** the effort to comprehend software requirements.

[2] lglwww.epfl.ch/research/use_cases/

The objective of the statistical analysis will be rejecting the above null hypotheses and possibly accepting the alternative hypotheses, which can be easily derived.

These hypotheses are the same as we have investigated in the original experiment. Additionally, we are also interested in finding out whether screen mockups can play an important role as sources of information to perform the comprehension tasks. For this reason, we added to the comprehension questionnaire some questions to collect the source of information used by the students to answer (see Sec. 3.2.4). In particular we consider two level of importance: **Relevant** (mockups are more important than the average source of information) and **Predominant** (mockups are the first most important source of information). The membership of these two levels can be expressed by means of the following null hypotheses that were not investigated in the original experiment:

$H_{\mathbf{SR}0}$ The proportion of questions where screen mockups is the source of information used to answer **is equal or lower than** the average proportion of information sources.

$H_{\mathbf{SP}0}$ The proportion of questions where screen mockups is the source of information used to answer **is equal or lower than** the second highest proportion of information sources.

When we can reject $H_{\mathrm{SR}0}$ for a source, then the source belongs to the Relevant category; whenever $H_{\mathrm{SP}0}$ can be rejected then the source belongs to the Predominant category.

3.2.3 Design of the Replication

As for the original experiment, we used the *counterbalanced experimental design* [28] (see Table 1). It ensures that each subject works on different *Objects* (AMICO or EasyCoin) in two *Tasks*, receiving each time a different *Treatment*, namely S (use cases and Screen mockups) and T[3] (use cases alone). This design has been also chosen because permits the use of statistical tests (e.g., two-way ANOVA) for studying the effect of other factors (e.g., Object) [28] and their interactions.

	Group 1	**Group 2**	**Group 3**	**Group 4**
Task 1	EasyCoin-S	EasyCoin-T	AMICO-T	AMICO-S
Task 2	AMICO-T	AMICO-S	EasyCoin-S	EasyCoin-T

Table 1: Experiment design

3.2.4 Selected Variables

In this replication the *control group* is Use Cases with no screen mockups while the *treatment group* is Use Cases augmented with mockups. Thus, the only one independent variable is Treatment, which is a nominal variable with two possible values: $\{S, T\}$.

The selected dependent variables are the *comprehension effort* required to complete the tasks and the *comprehension level* of the functional requirement.

[3]T is for Text only

The comprehension effort is measured as time to answer the questionnaire. It was recorded directly by the subjects noting down their start and stop time.

The comprehension level has been assessed, similarly to [17], using a *comprehension questionnaire* composed of 10 open questions on each software system. Questions are divided into three categories: *domain*, *IO*, and *development*. The *domain* category includes three questions concerning the domain of the system. The following four questions belong to the *IO* category and concern the interaction with the system that will be built (i.e., input, output) and the execution flow. Finally, *development* includes the remaining three questions, which concern implementation/development of the system. For instance, the third question of the comprehension questionnaire of EasyCoin is: "*A coin in EasyCoin contains the following information: beauty level and coin state. Report an example of coin state.*". As suggested in [5], we also collected data on the source of information the subjects used to answer the questions. To this end, we added an additional question for each question: "*Did the answer come from the Glossary (G), previous Knowledge (K), Screen mockups (S), Use Case (UC) or Use Case Diagrams (UCD)?*".

According to the defined comprehension questionnaires and similarly to [20], the comprehension level of the subjects has been assessed, using an information retrieval based approach. Since the plausible answers to each question consist of a list of string items (e.g., value, country, address) we can define as: $A_{s,i}$, the set of items mentioned in the answer to question i by subject s and C_i, the known correct set of items expected for question i.

We can then compute *precision* and *recall* [10] on each answer. In particular, precision measures the fraction of items in the answer that are correct, while recall measures the fraction of correct items that are in the answer:

$$precision_{s,i} = \frac{|A_{s,i} \cap C_i|}{|A_{s,i}|} \quad recall_{s,i} = \frac{|A_{s,i} \cap C_i|}{|C_i|}$$

Since precision and recall measure two different concepts (i.e., correctness and completeness), we used an aggregate measure to get a balance between them. In particular, we used the harmonic mean between these metrics [10], which assumes values between 0 and 1 and is defined as:

$$F{-}Measure_{s,i} = \frac{2 \cdot precision_{s,i} \cdot recall_{s,i}}{precision_{s,i} + recall_{s,i}}$$

To obtain a single measure of the comprehension level achieved by each subject, we computed the overall average of the F-Measure values of all the questions. In practice, a value close to 1 means that the student answered to the comprehension questionnaire very well, a value close to 0 means very bad.

The relevance of a source of information to perform the comprehension tasks can be measured as the proportion of subjects using it to answer a question. A source of average importance can be expected to be used by N/n_s subjects, where n_s is the number of alternative sources and N the number of subjects.

Additional variables that are measured are the Object ($\in \{AMICO, EasyCoin\}$), the Task ($\in \{Task1, Task2\}$) and

the Question category ($\in \{domain, IO, development\}$). The effect of these latter variables will be analyzed to get a more precise view of the results.

3.2.5 Execution and material

A pilot experiment was accomplished some days before the original experiment. The results [21] indicated that the experiment was well suited for undergraduate students. From it, and from the mean time of the original experiment to accomplish the tasks (104 minutes), we deduced that 2.5 hours in total without a break (or 3 hours with a break) were sufficient also for the subjects involved in the replication.

To carry out the replication we provided each subject with a computer. To execute the tasks, the subjects were asked to use the following procedure (we did the same in the original experiment): *(i)* specifying name and start-time in the comprehension questionnaire; *(ii)* answering independently the questions consulting the Use Cases; *(iii)* marking the end-time of the task in the comprehension questionnaire.

We did not suggest any approach on how facing the comprehension tasks. We only discouraged reading completely the Use Cases (they are too long). Note that the subjects could also use the Internet to get information.

To perform the experiment the subjects were provided with the following material:

- a MS Word® file of the requirements document. We chose an electronic format to permit the "Find" facilities that is very convenient with large documents;
- a paper copy of the comprehension questionnaires to be filled in;
- a paper copy of a unique post-experiment questionnaire to be filled in after the two tasks.

The post-experiment questionnaire aimed at gaining insights about the subjects' behavior during the experiment and at better explaining the obtained quantitative results. The questionnaire is composed of seven questions. A first group of questions (**Q1** through **Q5**) concerns the availability of sufficient time to complete the tasks, the clarity of the Use Case, and the ability of subjects to understand them. **Q6** is devoted to measure the perceived usefulness of the screen mockups, while **Q7** aims at measuring how much time, in percentage intervals, was spent to analyze Use Cases and screen mockups. All the questions, except **Q7** that is expressed in intervals of percentages, expected closed answers according to a five point Likert scale [18]: (1) strongly agree, (2) agree, (3) neither agree nor disagree, (4) disagree, (5) strongly disagree.

3.2.6 Analysis Procedure

In all our statistical tests we decided (as is customary) to accept a probability of 5% of committing Type-I-error [28], i.e., of rejecting the null hypothesis when it is actually true.

Because of the sample sizes (51 subjects) and mostly non-normality of the data we adopted non-parametric tests to reject the null hypotheses. In particular we selected Mann-Whitney and Wilcoxon tests because they are very robust and sensitive. Moreover, we used one-tailed statistical tests due to the directionality of the hypotheses. We also performed paired analysis whenever possible.

While the statistical tests allows for checking the presence of significant differences, they do not provide any information about the magnitude of such a difference. Accordingly, we used the Cohen's "d" standardized difference between two groups [9]. Typically, it is considered negligible for $d < 0.2$, small for $0.2 \leq d < 0.5$, medium for $0.5 \leq d < 0.8$, and large for $d \geq 0.8$.

We assess the relevance and predominance of mockups as information source by testing the proportion of subjects using them (when present). We compared the proportion to the average proportion (for relevance) and to the second highest proportion (for predominance). In this case a proportion test was used [2].

Finally, we measured the effect of the other factors on the dependent variables, namely of *Task* (i.e., to evaluate a possible learning/fatigue effect), and *Object*, using a two-way Analysis of Variance (ANOVA). In particular two-way ANOVA permits analyzing possible interactions between the main factor (in our case the Treatment) and co-factors (in our case the Object and Task).

3.2.7 Differences between the experiments

Based on the experience from the original experiment, some slight modifications were made to improve the material and the data analysis:

- *Update of requirements documents and questionnaires.* Small changes were made to the requirement documents and to the questionnaires to remove some sources of possible confusion and mistakes. For example, we better describe questions Q8 and Q10 of AMICO and added, in the EasyCoin requirement document, some missing links to the mockups;
- *Modifications to data analysis.* We added two new hypotheses related to the sources of information and analyzed them by means of the proportion-test. Moreover, we added a mosaic plot to highlight the source of information used by subjects to answer and used interaction plots (when useful) to show interaction between factors.

Some differences have been deliberately introduced, others are a natural consequence of having changed the subjects (students in the two experiments have different knowledge/skill), and finally some others are mainly related to time constraints:

1. The subjects of the replication are third-year undergraduate students while subjects of the original experiment were second-year students; thus the subjects of the replication are more skilled than the others. Thus, we expected better results in terms of comprehension and effort.

2. The subjects did not attend a lesson before the laboratory runs to introduce the SWEED template. This was because the students received already, at the beginning of the course, a similar lesson. Moreover, they knew very well the template as it was previously used for the Use Cases of the EasyCooking project (a system able to help a chef in storing and handling recipes). That is the project the students have to implement this year in the Software Engineering course hosting the experiment.

3. An exercise similar to the tasks of the experiment was not conducted before the replication for the same motivation of the previous point; they already knew very well the notation used for presenting the Use Cases.

4. A half an hour break between the two tasks was not provided for time constraint (the laboratory was available only for 2.5 hours). We expected a bigger fatigue effect with respect to the original experiment.

5. In the original experiment we used the information gathered in a *pre-questionnaire* to equally distribute high and low ability/experience subjects among the groups [21]. Instead, in the replication, the subjects were not asked to fill in a such kind of questionnaire for time constraint. Therefore, the subjects were randomly assigned to the groups in Table 1.

4. ANALYSIS AND RESULTS

In the following, we first introduce the data from the original and replicated experiments, which are identified with the names *PZ* and *GE* respectively (i.e, the location where the experiment was conducted, PZ=Potenza and GE=Genova). Successively, we present the data analysis of the replication.

4.1 Comparison of the Results

To compare the results of the two experiments, the overall values of comprehension Level and Effort must be considered. Figure 3 presents the boxplots of the Effort and Comprehension Level for the two experiments (without partitioning by Treatment). The difference that can be observed graphically is confirmed by means of Mann-Whitey (MW) tests. Table 2 presents the summary statistics and the results of the tests. We can observe that the mean value of Comprehension Level in the second experiment (GE) is 7 points (13.2%[4]) higher than in the first experiment (PZ) ($p = 0.01$). As far as Comprehension Effort is concerned, again the mean effort in the second experiment is about 10 minutes (20.5%) lower than in the first one ($p = 8 \cdot 10^{-7}$). Given these differences, we cannot simply merge the data from the two experiments. As a consequence, the two data sets ought to be analyzed separately and then we can draw joint conclusions from the results.

4.2 Comprehension Level

Table 3 reports the main descriptive statistics of Comprehension Level for both experiments, divided by Object, and Question category. Focusing only on the second experiment (GE), the overall comparison (i.e., without partitioning by Object) is visually presented in Figure 4 (left) by means of boxplots. From them, it is apparent that students with screen mockups (S) outperform in comprehension students without them (T).

[4]The percentage comes from the following equation: 0.53+0.53*x%=0.60

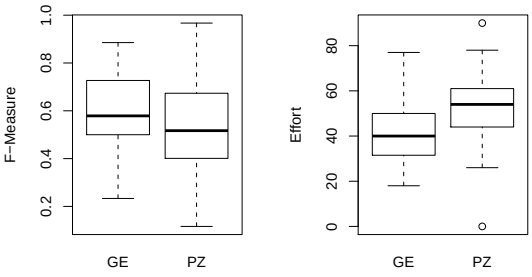

Figure 3: Boxplots of Comprehension and Effort.

Var	GE mean	GE med	GE σ	PZ mean	PZ med	PZ σ	p
Level	0.60	0.58	0.15	0.53	0.52	0.19	**0.01**
Effort	41.25	40.00	13.15	51.86	54.00	14.38	**<0.01**

Table 2: Descriptive statistics without partitioning by Treatment.

Exp	Object		S mean	S median	S σ	T mean	T median	T σ	p
GE			0.67	0.67	0.15	0.53	0.53	0.12	**<0.01**
	AMICO		0.76	1.00	0.34	0.52	0.50	0.39	**<0.01**
		domain	0.89	1.00	0.28	0.58	0.74	0.43	**<0.01**
		IO	0.81	1.00	0.31	0.52	0.64	0.41	**<0.01**
		develop	0.55	0.50	0.33	0.47	0.50	0.30	0.05
	EasyCoin		0.59	0.67	0.44	0.55	0.67	0.43	0.09
		domain	0.78	1.00	0.36	0.79	1.00	0.33	0.20
		IO	0.71	0.97	0.39	0.60	0.67	0.41	**0.01**
		develop	0.22	0.00	0.36	0.25	0.00	0.38	0.67
PZ			0.63	0.67	0.17	0.43	0.45	0.14	**<0.01**
	AMICO		0.64	1.00	0.43	0.40	0.31	0.43	**<0.01**
		domain	0.79	1.00	0.38	0.37	0.00	0.44	**<0.01**
		IO	0.64	1.00	0.44	0.37	0.00	0.45	**<0.01**
		develop	0.48	0.50	0.41	0.47	0.50	0.40	0.47
	EasyCoin		0.62	0.75	0.42	0.46	0.50	0.44	**<0.01**
		domain	0.81	1.00	0.34	0.68	0.86	0.40	**0.02**
		IO	0.64	0.71	0.42	0.51	0.67	0.43	**0.04**
		develop	0.39	0.50	0.40	0.19	0.00	0.33	**<0.01**

Table 3: Descriptive statistics of Comprehension Level and the results of MW tests.

We evaluate the first hypothesis overall. Both paired ($p = 2 \cdot 10^{-6}$) and unpaired ($p = 1 \cdot 10^{-5}$) Mann-Whitney tests provide evidence that there exists a significant difference in terms of Comprehension Level as effect of the presence of screen mockups. Therefore, in general, we can reject the null hypothesis H_{l0}. From a practical point of view, the difference can be considered large ($d = 0.97$). The mean Level improvement achieved with screen mockups is of 14 points (26.4%).

Focusing on the separate objects we can observe a significant difference for AMICO ($p < 0.01$) and no significant difference for EasyCoin ($p = 0.09$). Detailing further on

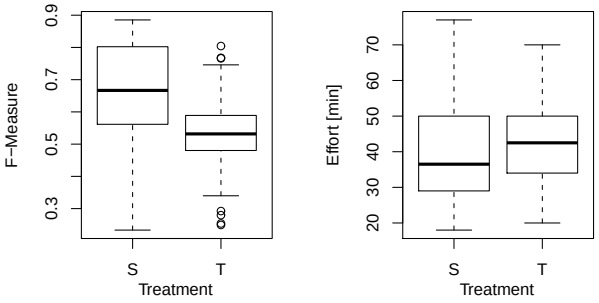

Figure 4: Boxplots of Comprehension Level and Effort by Treatment for the two experiments.

Question Categories, it appears that for the former application the difference exists for the domain and IO category. Instead for the latter, we found a significant difference for the IO category only ($p = 0.01$). The first null hypothesis, as far as specific Objects are concerned, can be rejected for AMICO but not for EasyCoin. The practical difference is extremely large ($d = 1.7$) for AMICO and small ($d = 0.32$) for EasyCoin.

4.3 Comprehension Effort

Table 4 reports the main descriptive statistics of Comprehension Effort for both experiments, divided by Object. The overall comparison of the second experiment (GE) is presented graphically in Figure 4 (right).

We observe no statistically significant difference ($p = 0.1$) in terms of comprehension effort, therefore we cannot reject the null hypothesis H_{e0} in general. The practical difference is small ($d = -0.2$). Even analyzing the two *Objects* separately no significant difference was found ($p = 0.1$ for AMICO and $p = 0.11$ for EasyCoin), even the effect size is small ($d = -0.3$ for both Objects). Therefore we cannot reject hypothesis H_{e0} for either Objects.

Exp	Object	S			T			
		mean	med	σ	mean	med	σ	p
GE	All	39.90	36.50	13.80	42.60	42.50	12.46	0.10
	AMICO	49.08	50.00	12.34	52.28	50.00	8.64	0.10
	EasyCoin	30.72	33.50	7.80	32.92	34.00	6.88	0.11
PZ	All	51.12	54.00	14.48	52.61	55.00	14.45	0.19
	AMICO	60.47	60.00	11.34	55.75	60.00	16.97	0.61
	EasyCoin	41.19	40.00	10.30	49.65	49.00	11.34	**0.02**

Table 4: Descriptive statistics of Comprehension Effort and the results of MW tests.

4.4 Source of information

Figure 5 presents a mosaic plot reporting the frequency of the sources of information used by the students in the second experiment (GE) to answer the questions by Question Category. Highlighted (in green for colored screen) are the screen mockups. From that plot it is evident the relevance of the screen mockups (see columns "S") during the requirements understanding phase.

Table 5 presents the details about the proportion of subjects who used screen mockups as their primary source of information in answering the comprehension questions. The Table reports the number of questions considered (N), the aforementioned proportion, and the p-values of the test relative to the Relevant (H_{SR0}) and Predominant (H_{SP0}) classes. The results are presented by Object and Question category.

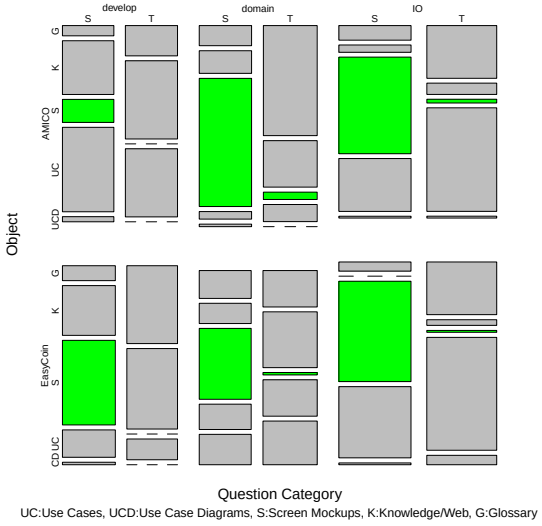

Figure 5: Mosaic plot of sources of information by Question Category and Object.

We observe that in general Screen mockups represent a both relevant and predominant source of information, i.e., at the general level we can reject both H_{SR0} and H_{SP0}. On average 48% of the questions were answered resorting to the information conveyed by screen mockups. More precisely, this holds in all case except the development Question Category of Object AMICO.

Object	Category	N	Prop.	Relevant p	Predominant p
All	All	477	0.48	**<0.01**	**<0.01**
AMICO	All	236	0.48	**<0.01**	**<0.01**
	domain	72	0.71	**<0.01**	**<0.01**
	IO	95	0.56	**<0.01**	**<0.01**
	develop	69	0.13	0.90	1.00
EasyCoin	All	241	0.49	**<0.01**	**<0.01**
	domain	69	0.41	**<0.01**	**<0.01**
	IO	100	0.55	**<0.01**	**<0.01**
	develop	72	0.47	**<0.01**	**<0.01**

Table 5: Mockup source proportion and test results, by Object and Question Category

4.5 Co-factors

The two co-factors we analyze are *Object* and *Task*. The former lets us evaluate whether system characteristics or familiarity with the application domain influence, in some way, the benefits deriving from screen mockups. The latter accounts for learning or fatigue effects. We analyze the effects of co-factors on both Comprehension Level and Effort.

Object: Two-way ANOVA reveals a statistically significant interaction between Object and Treatment (see Table 6) con-

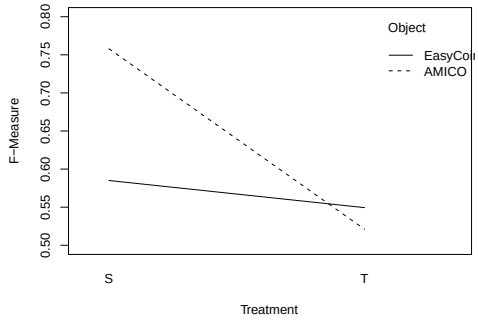

Figure 6: Interaction plot of Treatment and Object vs. Level.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	1	0.46	0.46	29.24	**<0.01**
Object	1	0.13	0.13	8.40	**0.0046**
Treatment:Object	1	0.26	0.26	16.29	**0.0001**
Residuals	98	1.55	0.02		

Table 6: Two-way ANOVA of Treatment and Object vs. Comprehension Level

firmed also by the interaction plot of Figure 6. In addition we observe a significant effect ($p = 0.0046$) of the co-factor Object by itself. The effect size is small ($d = 0.46$). Instead, from a two-way ANOVA analysis of Treatment and Object vs. Effort (see Table 7), we can observe a significant effect of Object ($p < 0.01$) but no interaction with the treatment. The effect size is quite large ($d = 2.06$) having AMICO a mean effort of 50 minutes and EasyCoin a mean effort of 32 minutes.

Task: The results from a two-way ANOVA analysis of Comprehension Level by Treatment and Task shows that Task both has a significant effect ($p = 0.0092$) and interacts with Treatment ($p < 0.01$). Specifically in the second task we observe a consistent reduction of the effect of the treatment (-80.4%). As far as the Comprehension Effort is concerned, we observe also a significant effect of Task ($p < 0.01$), while no interaction with the treatment was revealed ($p = 0.41$). In the second task we observe a reduction of the mean time to complete it (-35.3%).

4.6 Post Questionnaire Results

From the post questionnaire analysis we know that subjects had sufficient time to complete the required tasks (PQ1 median = 1, i.e. Fully Agree). In general the subjects deemed the tasks clear and useful (PQ2 through PQ5 median = 2, i.e. Agree). The subjects fully acknowledged (PQ6 median = 1) the screen mockups usefulness. Moreover they report spent 30% (median value) of the time examining the proposed screen mockups.

5. DISCUSSION

In the following we discuss the issues related to the execution of this replication and the obtained results.

5.1 Documentation and Communication Issues

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	1	182.25	182.25	2.18	0.1435
Object	1	8892.49	8892.49	106.13	**<0.01**
Treatment:Object	1	6.25	6.25	0.07	0.7854
Residuals	96	8043.76	83.79		

Table 7: Two-way ANOVA of Treatment and Object vs. Comprehension Effort

The success of a replication may be influenced by issues such as documentation and communication between experimenters [23, 27]. As the language of the subjects involved in the replication was the same of the subjects of the original experiment, the issues concerning the translation of the experimental material have been mitigated. Moreover, the two groups of experimenters frequently conducted call conferences using Skype and exchanged e-mails to share knowledge on the original experiment in order to reproduce as closely as possible the same setting as the one in the original experiment. Subversion was also used to share and record the evolution of the adopted experimental material.

5.2 Discussion on the results

Globally, the results confirm the findings of the original experiment with only slightly differences, thus increasing our awareness on the benefit deriving from the use of screen mockups. As in the first experiment, *(i)* the null hypothesis H_{l0} can be rejected with high confidence, *(ii)* screen mockups represent a both relevant and predominant source of information (we can reject both H_{SR0} and H_{SP0}) and, *(iii)* students fully acknowledged the screen mockups usefulness and effectiveness in Requirements understanding (PQ6 median = 1).

Even if, the data confirm (as expected) that the students of the replication are more skilled than the previous ones (results in terms of overall average F-measure and effort are better), the standardized effect size of the treatment on the comprehension level decreased with respect to the original experiment (from $d = 1.2$ to $d = 0.97$). We observe the same difference also in terms of mean comprehension level improvement: +26.4% improvement with mockups in the replication vs. +46.5% of the original experiment. Moreover, in the replication we observed a more relevant difference between the two objects that needs further investigation. In the original experiment a significant improvement of the treatment was found for both applications, although the improvement for AMICO was slightly larger than for EasyCoin. Instead, in the replication the improvement is extremely large for AMICO and small for EasyCoin. Given that, the domain of EasyCoin (i.e., coin collection) is more popular and the subjects were less familiar with the domain of AMICO (condominium management), these results seem to confirm and reinforce that the degree of benefits attainable from the presence of screen mockups appear to be inversely proportional to the familiarity.

When focusing on specific comprehension categories, we expected the *IO* and *domain* Question category to benefit more from screen mockups, while the development Question category was expected to be affected less because mockups are mostly related with the user interface and scarcely, if

not at all, with internal implementation details. In practice, we observed that the only non-significant differences were found for the *development* and *domain* Question category in EasyCoin, and for *development* category in AMICO (really *development* is only marginally significant). Similar results were obtained in the original experiment.

Similarly to the original experiment, the results of the replication confirmed no substantial variation of effort to comprehend software requirements, whether subjects are provided or not with screen mockups. Actually, a reduction of effort is observed when mockups are present but the difference is not statistically significant. Thus, as in the first experiment, we cannot reject globally the null hypothesis H_{e0}. Even analyzing the two *Object*s separately no significant difference was found (instead in the original experiment we rejected H_{e0} for EasyCoin). In contrast with the original experiment where it seemed that *familiarity* with the application domain affected positively the benefits achieved by adding mockups, here that *familiarity* does't affect the benefits of mockups on effort.

Clearly, the influence of the factor "familiarity" on comprehension and effort needs further investigations. Future replications with different objects are planned to better understand that aspect.

5.3 Threats to Validity

Internal validity threats concerns factors that may affect a dependent variable. They can be due to learning and fatigue effects. Even if the learning effect should be mitigated by the chosen experiment design, we found a significant effect of *Task* on the comprehension level (ANOVA $p = 0.0092$) and on the effort (ANOVA $p < 0.01$). Moreover, we observed a significant interaction between *Task* and the main factor on the comprehension level (ANOVA $p = 0.0004$), while no interaction on the Effort was revealed (ANOVA $p = 0.41$). Specifically, in the second task we observed a consistent reduction of the effect of the treatment on the comprehension level and a reduction of the mean time to complete it. These results could be the consequence of having removed the break between the two tasks and eliminated the *LaTazza* exercise before the experiment. Indeed, in the original experiment, where the break was imposed and the exercise done, these effects were lighter. In additon, to avoid apprehension, students were not evaluated on their performance. They were also not aware of the experimental hypotheses.

External validity concerns the generalization of the findings. Threats belonging to this category are mainly related to the tasks and objects of the experiment and to the use of students as experimental subjects. Concerning the first point, we can argue that requirements documents of AMICO and EasyCoin are realistic for small size projects and they are not trivial. Tasks have been designed simple to fit the time constraints of the experiment (2.5 hours). Concerning the second point, we can say that the selected subjects represent a population of students specifically trained on software engineering tasks and in particular on requirements. This makes these subjects not so much inferior to professional young junior developers. Moreover, Kitchenham et al. [16] argue that using students as subjects instead of software engineers is not a major issue, as long as the research questions are not specifically focused on experts. In addition, it has been shown, in a specific context of requirements engineering, that students have a good understanding of the way industry acts, and may work well as subjects in empirical studies in this area [26]. Clearly, further studies with larger/real objects, more demanding tasks and more experienced subjects are needed to confirm or contradict the obtained results.

Construct validity threats concern the relationship between theory and observation. This threat is related to how comprehension level and effort were measured. Comprehension was measured using questionnaires and answers were evaluated using an information retrieval based approach in order to avoid as much as possible any subjective evaluation. The same approach was also used in other studies (e.g., [17, 20]). Concerning the questionnaire, we carefully defined the questions so that they were neither too complicated to make the tasks impossible to perform nor too simple to make it difficult to observe any difference among subjects. Effort was measured by means of proper time sheets and validated qualitatively by researchers, who were present during the experiment. Although this may not be very accurate, this is a widely adopted way of measuring effort.

Conclusion validity concerns the relationship between the treatment and the findings. In our study non-parametric tests (Mann-Whitney test for unpaired analyses, the Wilcoxon test for paired analyses) were performed to statistically reject the null hypotheses, while two-way ANOVA was used to detect possible interactions between each co-factor and the main factor. Even if all the assumptions/conditions for using ANOVA were not checked (e.g., ANOVA assumes distributions symmetric and homogeneous variance between cells) this test is quite robust and has been used extensively in the literature to conduct analysis similar to ours. The post-questionnaires were designed using standard ways and scales [18].

6. CONCLUSIONS

We have presented an internal replication of an experiment for assessing the effectiveness of including screen mockups when adopting Use Cases. Our results confirms the findings of the original experiment with only slight, almost negligible differences. They indicate a clear improvement in the comprehension of software requirements when screen mockups are present with no significant impact on effort.

Future replications will aim at investigating: *(i)* the effects of changing the domain and the complexity of the tasks; *(ii)* the motivation of the observed difference between objects (EasyCoin and AMICO) in the comprehension level; *(iii)* the influence of the factor "familiarity"; *(iv)* whether benefits of screen mockups will keep also for other categories of subjects (e.g., graduated students, Ph.D. students, and professional software engineers). In addition, it would be worth analyzing whether the additional effort and cost, due to the development of screen mockups, will be paid back by a improved comprehension of Use Cases. Indeed, from a manager point of view, the adoption of screen mockups, as prototyping tool in the software development life-cycle, should take into account the costs it will introduce.

7. ACKNOWLEDGMENTS

We would like to thank the students that took part in the original and replicated experiments.

8. REFERENCES

[1] S. M. Abrahão, E. Insfrán, C. Gravino, and G. Scanniello. On the effectiveness of dynamic modeling in UML: Results from an external replication. In *Symposium on Empirical Software Engineering and Measurement*, pages 468–472, Lake Buena Vista, FL, USA, 2009. IEEE Computer Society.

[2] A. Agresti. *An Introduction to Categorical Data Analysis*. Wiley-Interscience, 2007.

[3] B. Anda, D. I. K. Sjøberg, and M. Jørgensen. Quality and understandability of use case models. In *European Conference on Object-Oriented Programming*, pages 402–428, London, UK, 2001. Springer-Verlag.

[4] G. Andrew and P. Drew. Use case diagrams in support of use case modeling: Deriving understanding from the picture. *Journal of Database Management*, 20(1), 2009.

[5] J. Aranda, N. Ernst, J. Horkoff, and S. Easterbrook. A framework for empirical evaluation of model comprehensibility. In *Modeling in Software Engineering, ICSE Workshop*, pages 7–13. IEEE, 2007.

[6] V. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Trans. Softw. Eng.*, 25(4):456–473, 1999.

[7] L. Bratthall and C. Wohlin. Is it possible to decorate graphical software design and architecture models with qualitative information?-An experiment. *IEEE Trans. Softw. Eng.*, 28(12):1181–1193, 2002.

[8] B. H. C. Cheng and J. M. Atlee. Research directions in requirements engineering. In *Future of Software Engineering*, pages 285–303. IEEE, 2007.

[9] J. Cohen. *Statistical power analysis for the behavioral sciences (2nd ed.)*. Lawrence Earlbaum Associates, Hillsdale, NJ, 1988.

[10] W. B. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[11] C. Gravino, G. Scanniello, and G. Tortora. An empirical investigation on dynamic modeling in requirements engineering. In *Conference on Model Driven Engineering Languages and Systems*, pages 615–629, Berlin, Heidelberg, 2008. IEEE Computer Society.

[12] D. Hendrix, J. Cross, and S. Maghsoodloo. The effectiveness of control structure diagrams in source code comprehension activities. *IEEE Trans. Softw. Eng.*, 28:463–477, 2002.

[13] M. Jackson. *Software Requirements And Specifications (ACM Press Books)*. Addison-Wesley Professional, August 1995.

[14] N. Juristo and A. Moreno. *Basics of Software Engineering Experimentation*. Kluwer Academic Publishers, Englewood Cliffs, NJ, 2001.

[15] N. Juristo and S. Vegas. Using differences among replications of software engineering experiments to gain knowledge. In *International Symposium on Empirical Software Engineering and Measurement*, pages 356–366, Lake Buena Vista, FL, USA, 2009. IEEE Computer Society.

[16] B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin, K. El Emam, and J. Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Softw. Eng.*, 28(8):721–734, 2002.

[17] L. Kuzniarz, M. Staron, and C. Wohlin. An empirical study on using stereotypes to improve understanding of UML models. In *Workshop on Program Comprehension*, pages 14–23, Bari, Italy, 2004. IEEE Computer Society.

[18] A. N. Oppenheim. *Questionnaire Design, Interviewing and Attitude Measurement*. Pinter, London, 1992.

[19] M. C. Otero and J. J. Dolado. An empirical comparison of the dynamic modeling in OML and UML. *Journal on Systems and Software*, 77(2):91–102, 2005.

[20] F. Ricca, M. Di Penta, M. Torchiano, P. Tonella, and M. Ceccato. The role of experience and ability in comprehension tasks supported by UML stereotypes. In *International Conference on Software Engineering*, pages 375–384, Minneapolis, MN, USA, May 2007. IEEE Computer Society.

[21] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano. Can screen mockups improve the comprehension of functional requirements? In *http://www.scienzemfn.unisa.it/scanniello/ScreenMockupExp/*, 2010.

[22] F. Ricca, M. Torchiano, M. Di Penta, M. Ceccato, and P. Tonella. Using acceptance tests as a support for clarifying requirements: a series of experiments. *Information & Software Technology*, 51(2):270 – 283, 2009.

[23] F. Shull, M. Mendonça, V. Basili, J. Carver, J. C. Maldonado, S. Fabbri, G. Travassos, and M. Ferreira. Knowledge-sharing issues in experimental software engineering. *Empirical Software Engineering*, 9(1-2):111–137, March 2004.

[24] D. I. K. Sjoberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N. Liborg, and A. C. Rekdal. A survey of controlled experiments in software engineering. *IEEE Trans. Softw. Eng.*, 31(9):733–753, 2005.

[25] M. Staron, L. Kuzniarz, and C. Wohlin. Empirical assessment of using stereotypes to improve comprehension of uml models: A set of experiments. *Journal of Systems and Software*, 79(5):727–742, 2006.

[26] M. Svahnberg, A. Aurum, and C. Wohlin. Using students as subjects - an empirical evaluation. In *Symposium on Empirical Software Engineering and Measurement*, pages 288–290, Kaiserslautern, Germany, 2008. IEEE Computer Society.

[27] S. Vegas, N. J. Juzgado, A. M. Moreno, M. Solari, and P. Letelier. Analysis of the influence of communication between researchers on experiment replication. In *International Symposium on Empirical Software Engineering*, pages 28–37, Rio de Janeiro, Brazil, 2006. IEEE Computer Society.

[28] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering - An Introduction*. Kluwer, 2000.