A Controlled Experiment for Assessing the Contribution of Design Pattern Documentation on Software Maintenance

Giuseppe Scanniello  
Dipartimento di Matematica e Informatica, University of Basilicata  
Viale dell’ Ateneo, I-85100, Italy  
+39 0971205881  
giuseppe.scanniello@unibas.it

Carmine Gravino, Michele Risi, Genoveffa Tortora  
Dipartimento di Matematica e Informatica, University of Salerno  
via Ponte don Melillo, I-84084, Italy  
+39 097963503  
{gravino, mrisi, tortora}@unisa.it

ABSTRACT

In this paper we present the preliminary results of a controlled experiment to assess the contribution provided by the design patterns on the maintenance of source code. In particular, the study aimed at assessing the effort and the efficiency to perform maintenance operations in case design pattern instances are properly documented and provided to the maintainer. The context of the experiment is constituted of Master Students in Computer Science at the University of Basilicata. The preliminary analysis conducted on the gathered data revealed that the effort is significantly reduced in case design pattern instances are properly documented and provided to the subjects. Similarly, the efficiency is significantly better in case the documentation of design pattern instances is used to accomplish maintenance operations.

Categories and Subject Descriptors

D.2.0 [Software Engineering]: General

General Terms

Documentation, Design, Experimentation, Human Factors.

Keywords

Design Pattern, Controlled Experiment, Maintenance.

1. INTRODUCTION

The maintenance process is needed to ensure that a software system continues to satisfy users’ requirements. This relevant phase is applicable to software systems developed by using any software life cycle model and programming language. Swanson [5] identified three kinds of changes that can be performed to let a software system evolve during the time: (i) corrective; (ii) perfective; (iii) adaptive. A corrective maintenance operation is a reactive modification to correct discovered problems, while a perfective maintenance operation is need to improve the performance or the maintainability of an existing software system.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EXPERIMENT DESIGN

In this section we present the design of the controlled experiment following the guidelines proposed by Wohlin et al. [6]. For replication purposes the experimental package and the gathered data are available at www.scienzemfn.unisa.it/scanniello/DP_1.
2.1 Context
The experiment has been conducted within a laboratory of the University of Basilicata (Italy) with 24 students of the Master program in computer Science. The experiment represented an optional activity of an Advanced Software Engineering course. Before accomplishing the experiment, the subjects had passed the exams of the following courses: basic and advanced object oriented programming languages, database system modeling, and software engineering.

All the subjects involved in the experiment were asked to accomplish an experimental task on the same source code. This was a chunk of a well known open source Java software system, namely JHotDraw (v5.1). The original comment of the selected classes was translated from the English language to the Italian language. This was needed to not bias the results since different subjects may have different familiarity with the English language. We also removed from the source code comment any reference to the implemented design patterns.

Some descriptive statistics on the task used in the experiment are shown in Table 1. It is worth mentioning that one of the authors manually detected the design patterns implemented among the selected classes. To accomplish this task the documentation of JHotDraw 5.1 and the PMAR1 data set have been properly used as well.

<table>
<thead>
<tr>
<th>LOCs</th>
<th>1326</th>
</tr>
</thead>
<tbody>
<tr>
<td># Classes</td>
<td>26</td>
</tr>
<tr>
<td># Line of comments</td>
<td>823</td>
</tr>
<tr>
<td># State Pattern</td>
<td>2</td>
</tr>
<tr>
<td># Adapter Pattern</td>
<td>1</td>
</tr>
<tr>
<td># Strategy Pattern</td>
<td>1</td>
</tr>
<tr>
<td># Decorator Pattern</td>
<td>1</td>
</tr>
<tr>
<td># Composite Pattern</td>
<td>1</td>
</tr>
<tr>
<td># Observer Pattern</td>
<td>1</td>
</tr>
<tr>
<td># Command Pattern</td>
<td>1</td>
</tr>
<tr>
<td># Template Method Pattern</td>
<td>1</td>
</tr>
<tr>
<td># Prototype Pattern</td>
<td>1</td>
</tr>
</tbody>
</table>

2.2 Hypotheses Formulation
In this study we are interested in the design patterns as a mean to improve the maintenance of existing object oriented software systems. Accordingly, the perspective of this study is from the point of view of the Researchers, investigating the effort and the efficiency related to the presence and the documentation of design pattern instances to perform maintenance tasks, and of the Project Managers, evaluating the possibility of properly documenting design patterns for the maintenance of their software systems. We have then investigated the following one-tailed null hypotheses:

**H0.** The documentation of design pattern instances does not significantly reduce the effort to perform maintenance operations on source code.

**H1.** The documentation of design pattern instances does not significantly increase the efficiency in performing maintenance operations on source code.

The goal of the statistical analysis is to reject the defined null hypotheses and possibly accepting the alternative ones (i.e., Ha0 and Ha1), which can be easily derived.

2.3 Experiment Design
We used a completely randomized design [6]. The design set up uses the same Object (i.e., JHotDraw) for two treatments, namely DP (Design Patterns) and NO_DP (NO Design Patterns). The subjects are randomly assigned to each treatment. We assigned the same number of subjects (i.e., 12) to each treatment, thus balancing the design. Each subject used only one treatment, namely either DP or NO_DP. This design has been chosen as we were interested in assessing the effect of using properly documenting design pattern instances (i.e., the main factor of our study) on the execution of small corrective and perfective maintenance operations. It is also worth noting that we were interested in using a realistic task (as much as possible) both in terms of size and complexity, thus mitigating external validity threats. This did not enable the use of experiment design, such as the counterbalanced design, for studying the effect of other factors. The use of different experiment designs with a non trivial Object may threaten the validity of the results introducing a factor difficult to be controlled, e.g., the mental fatigue.

2.4 Selected Variables
The control group is the source code with no documented design pattern instances, while the treatment group is the source code with the documentation of the design pattern instances. Thus, the only one independent variable is Treatment (i.e., the main factor), which is a nominal variable with two possible values: DP and NO_DP.

Two are the selected dependent variables: Efficiency. The former is measured as the time (expressed in minutes) needed to accomplish the task. This was recorded directly by each subject noting down his/her start and stop time.

On the other hand, Efficiency is computed dividing a quantitative evaluation of the subject performance by Effort. Subjects’ performances have been evaluated through a questionnaire composed of 14 open questions. This questionnaire aimed at assessing the ability of the subjects to perform corrective and perfective maintenance operations. We also introduced questions to get an indication on the subjects’ comprehension level of the source code. The questions have been divided in three parts to let the subject take a break if needed when passing from a part to another. Each taken break was controlled by the supervisors to avoid the possible exchange of information among the subjects.

To quantitatively evaluate subjects’ performances we adopted an information retrieval based approach [4], where the correctness and completeness of each answer has been assessed using precision and recall, respectively. These measures are defined as:

\[
\text{precision}_{i} = \frac{\text{correct}_{i}}{\text{answer}_{i}}
\]

\[
\text{recall}_{i} = \frac{\text{correct}_{i}}{\text{correct}}
\]

\text{answer}_{i} was the set of answers that the subject \( s \) provided for the question \( i \), while \text{correct}_{i} was the set of correct answers expected for the question \( i \). To get a balance between precision and recall, we used the harmonic mean of these measures. The overall average of the F-measures of all the questions was computed to get a quantitative indication of the result of each subject.
2.5 Preparation, Execution, and Data Analysis

A pilot experiment was accomplished some days before the experiment with a research fellow and a PhD student. The results indicated that the experiment was well suited for Master students. This also showed some minor issues in the experimental material that were properly addressed before conducting the experiment.

A training session was accomplished by the subjects some days before the actual experiment. In particular, an exercise similar to the experimental task was accomplished. The subjects were also asked to fill in a pre-questionnaire to gather information about passed exams, industrial working experience, and grade point average.

To carry out the experiment each subject was provided with a computer, where a software tool was installed to surf the source code of the task. They were asked to use the following procedure for each part of the questionnaire: (i) specifying name and start-time; (ii) answering the questions surfing the source code; (iii) marking the end-time. We did not suggest any approach to accomplish the task. We only discouraged surfing completely the source code.

In order to perform the experiment, a paper copy of the following experimental material was provided to all the subjects: (i) the questionnaire to assess the subjects’ performances; (ii) a post-experiment survey questionnaire. Additionally, the subjects that experimented DP were provided with the paper copy of: (iii) the class diagram of the chunk of the considered software system; (iv) a document where the design pattern instances were reported on.

It is worth noting that the goal of the post-experiment survey questionnaire was to gain insight in order to better explain the results. Two different questionnaires were defined for DP and NO_DP, respectively.

Regarding the questionnaire to assess subjects’ performances, we also collected data on the source of information used for answering the questions. In particular, the subjects that accomplished the task employing DP were asked to specify for each question whether the answer was derived by using: design pattern instances, class diagram, previous knowledge, source code comments, or source code. The other subjects were asked to choose between: source code comments and source code.

To investigate the defined null hypotheses we adopted non-parametric tests due to the sample size and mostly non-normality of the data. In particular, we used the Mann-Whitney test because it is very robust and sensitive [6]. In all the performed statistical tests we decided (as custom) to accept a probability of 5% of committing Type-I-error [6], i.e., of rejecting the null hypothesis when it is actually true.

3. ANALYSIS AND DISCUSSION

Some descriptive statistics (i.e., median, mean, and standard deviation), grouped by Treatment, Effort, and Efficiency are shown in Table 2. The gathered data are also summarized by the boxplots in Figure 1.

3.1 Influence of the Main Factor

The results of the Mann-Whitney test are summarized in Table 3. In particular, it shows that the null hypothesis Hn0 and Hn1 can be rejected since the p-values were 0.003 and 0.008, respectively.

This means that the documentation of design pattern instances significantly reduces the effort to perform maintenance operations. On the other hand, the efficiency significantly increases when the presence of design pattern instances is properly documented and provided to the maintainers.

Table 2. Descriptive Statistics

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort</td>
<td>157</td>
<td>157.7</td>
<td>28.109</td>
<td>182</td>
<td>187.6</td>
<td>16.989</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0.320</td>
<td>0.312</td>
<td>0.044</td>
<td>0.230</td>
<td>0.248</td>
<td>0.037</td>
</tr>
</tbody>
</table>

Table 3. Mann-Whitney Results

<table>
<thead>
<tr>
<th>Null Hypotheses</th>
<th>p-value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hn0</td>
<td>Yes</td>
<td>0.003</td>
</tr>
<tr>
<td>Hn1</td>
<td>Yes</td>
<td>0.008</td>
</tr>
</tbody>
</table>

Figure 1. Boxplots of Effort and Efficiency

3.2 Post Experiment Survey Questionnaire

The analysis of the answers of the post experiment survey questionnaire revealed that the time needed to carry out the experiment was considered appropriate and the objectives of the experiment were mainly clear. Moreover, the subjects judged the difficulty of the task as medium/high. Finally, the subjects generally considered useful the design choice documented through the design pattern instances.

3.3 Discussion

The results of the preliminary analysis presented in this paper provide evidence that maintainers achieve better performances when design pattern instances are provided together with the source code. In fact, the subjects who accomplished the task with DP on average spent 16% less than the ones who accomplished the task with NO_DP. The average level of efficiency obtained by the subjects that used DP was 21% more than the subjects that experimented NO_DP. These results find a plausible explanation in the further information that design patterns provide. However, this opens two interesting issues from the methodological point of view: (i) the effort to properly document design pattern instances; (ii) the possible presence of an experience threshold to benefit from the documentation of design patterns; (iii) the additional effort to document design pattern instances is paid back by an improvement of the maintainers’ performances. Future work is needed to investigate these issues.

Regarding the distribution of the two selected dependent variables, the boxplots in Figure 1 show an interesting point. In particular, the length and tails of the boxplots of DP for both the dependent variable are more skewed, thus indicating a possible effect of the subjects’ ability to benefit from the documentation of design pattern instances. This also represents a further possible direction for our work.
Additional insights can be gathered by looking at which sources of information were used by the subjects. In particular, we observed that subjects achieved a better Efficiency value on the questions where they were able to correctly identify the design patterns involved to answer them. We do not provide further details here for space reasons.

4. THREATS TO VALIDITY

Internal validity threats are mitigated by the design of the experiment. Each group of subjects worked only on one task, with or without the design patterns instances. Another possible threat concerns the exchange of information among the subjects. We prevented this in several ways. Internal validity threats could be also due to fatigue effect. We tried to mitigate it allowing the subjects to optionally take a break. In addition, 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 results. These threats are mainly related to the complexity of the task and to the use of students as subjects. Regarding the first point we selected a part of an open software system large enough to be considered not excessively easy. However, in the selected task a few numbers of design pattern instances (i.e., 10) were present, thus threatening the achieved results. Also the kinds of design patterns (i.e., 9) implemented in the considered source code may affect the results.

On the use of students as subjects, we can say that they were specifically trained on software engineering and object oriented programming tasks. Therefore, we think that this makes the subjects not so much inferior to professional junior developers. Indeed, it could be also possible that the subjects are better trained on design patterns than many senior software professionals.

Replications however are needed with different subjects (e.g., professionals) to confirm or contradict our results. We also plan to conduct different empirical investigations (e.g., industrial case studies) with more complex tasks. The reader may also object to the fact that the way as the design pattern instances have been documented may have affected the results. Also, the use of the paper copy of the class diagram of the excerpt of the system considered within the task may be positively affected the performances of the subjects that employed DP. Therefore, we plan to conduct an experiment in a similar context to assess the effect of these possible confounding factors.

In this study, construct validity may be influenced by the metrics used to get a quantitative evaluation of Efficiency and by social threats. The employed metrics were widely used in the past with purposes similarly to ours [4]. Regarding the Efficiency variable, one of the authors not involved in the definition of the task built the questionnaire to be complex enough so as not to be obvious. Finally, social threats (e.g., evaluation apprehension) have been mitigated as students were not graded on the obtained results.

Conclusion validity concerns issues that affect the ability of drawing a correct conclusion. In our study proper statistical tests were used. In particular, a non-parametric test (i.e., Mann-Whitney test for unpaired analyses) was used to statistically reject the null hypotheses. Conclusion validity could be also affected by the observation number.

5. CONCLUSIONS & FUTURE WORK

In this paper, we have presented the results of a controlled experiment to assess whether the additional information provided by the design pattern instances better support software engineers to perform maintenance operations. In the past a few numbers of papers have been proposed with goals similarly to ours [2][3]. However, in [2] the contribution of using only one design pattern on software maintenance is considered, while the effect of documenting design pattern instances in the source code is investigated in [3].

The greater part of the subjects involved in this study had experience with design patterns since they used them in an advanced course on object oriented programming language. The preliminary analysis of the gathered data revealed that: (i) the documentation of design pattern instances significantly reduces the time for executing maintenance operations; (ii) the efficiency significantly increases when design pattern instances are documented and used to accomplish maintenance operations.

Future empirical studies will be conducted to investigate whether the benefits deriving from the use of documented design pattern instances will keep also for other categories of subjects. In addition, it would be also worth analyzing whether the additional effort and cost, due to the development with design patterns, will be paid back by a reduction of the effort to perform maintenance operations and by an improvement of the efficiency.

The decision of using controlled experiments relies on the fact that a number of confounding and uncontrollable factors could be present in an industrial context. In fact, it may be mainly impossible to control factors such as learning and/or fatigue effects and to select specific tasks. Although questions about the generalization of the results may be arisen, controlled experiments are suitable in early steps of an empirical investigation. Thus, we have planned to conduct further empirical investigations in more realistic settings in terms of academic/industrial case studies.

6. REFERENCE


