# Talking Tests: an Empirical Assessment of the Role of Fit Acceptance Tests in Clarifying Requirements

Filippo Ricca
Unità CINI at DISI[*], Genova, Italy
filippo.ricca@disi.unige.it

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

Mariano Ceccato
Fondazione Bruno Kessler—IRST, Trento, Italy
ceccato@itc.it

Paolo Tonella
Fondazione Bruno Kessler—IRST, Trento, Italy
tonella@itc.it

## ABSTRACT

The starting point for software evolution is usually a change request, expressing the new or updated requirements on the delivered system. The requirements specified in a change request document are often incomplete and inconsistent with the initial requirement document, as well as the implementation. Programmers working on the evolution of the software are often in trouble interpreting an under-specified change request document, resulting in code that does not meet the users' expectations and contains faults that can only be detected later through expensive testing activities.

In this paper, we investigate the role of acceptance tests to clarify the requirements used in software evolution iterations. In particular we focus on *Fit* tables, a way to express acceptance tests which simplifies their translation into executable test cases. We designed and ran an experiment to assess whether availability of *Fit* tables affects the level of understanding and the productivity in understanding the requirements. Results indicate that *Fit* tables significantly improve requirement understanding, but tend to involve additional effort.

## Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications; D.2.5 [**Software Engineering**]: Testing and Debugging

## General Terms

Experimentation

## Keywords

Empirical studies, acceptance testing, requirements, Fit tables.

## 1. INTRODUCTION

### 1.1 Motivation

Software evolution is triggered by change requests, which can have different origins. Often the user asks for changes to accommodate new functionalities or improved functionalities into the existing system (*perfective* maintenance). Errors detected during the execution of the system originate bug reports, which trigger *corrective* maintenance evolution. Other reasons for change include the improvement of the internal code structure (*preventive* maintenance) and the co-evolution of the system with a changing environment, including software layers as operating system, database, GUI, etc. (*adaptive* maintenance).

Regardless of the origin of a change request, critical to the entire software evolution process is the way such a request is expressed. Usage of the natural language and free formats to specify the requirements for change is highly inaccurate and error prone, but represents the current state of the practice. Even worse, while we can usually assume that some care is taken when writing down the initial requirements, the role of a clear specification of the change requests is usually undermined and change request documents tend to be little more than a bug report or a short description, under the assumption that a lot of information is obvious at this time of the project or is agreed verbally.

Data indicate that on average 85% of the defects are estimated to originate from inadequate requirements [11]. This figure tends to be even higher during software evolution, compared to initial development. Ambiguous, incomplete, wishful thinking, inconsistent, unusable, silent, over-specific or over-sized software requirements [5] are the main causes of software development and evolution problems, and eventually of faults.

Recent work in test-driven development emphasizes the contribution of testing and test cases, when their construction anticipates the actual development of the code. Tools like *JUnit*, which are often integrated into the software development environment, support early construction and au-

[*] Laboratorio Iniziativa Software FINMECCANICA/ELSAG spa - CINI

IWPSE'07, September 3-4, 2007, Dubrovnik, Croatia. Copyright 2007 ACM ISBN 978-1-59593-722-3/07/09...5.00.

tomated execution of test cases by developers. The same approach was recently moved to the previous phase of development and evolution, when the change requests coming from the user are collected. In this phase, test cases take the form of acceptance tests, in that they specify the expected behavior of the system from the point of view of the user (vs. the developer's point of view in unit testing).

The "agile movement" advocates that tests can effectively complement high-level requirements, constituting an expressive and precise form of documentation of the change requests. Test cases are considered to be able to express detailed requirements more precisely than natural language [4].

In this paper, we focus on Fit [6], one of the most popular methodologies that support the creation of acceptance tests as a support to clarify requirements. The goal of the present work is to assess the contribution of acceptance tests, used to clarify the requirements for change before actually implementing the change. In particular, we are interested in evaluating whether acceptance tests make requirement understanding easier and what is the involved cost, in terms of extra effort (if any) devoted to test case understanding.

## 1.2 What is Fit?

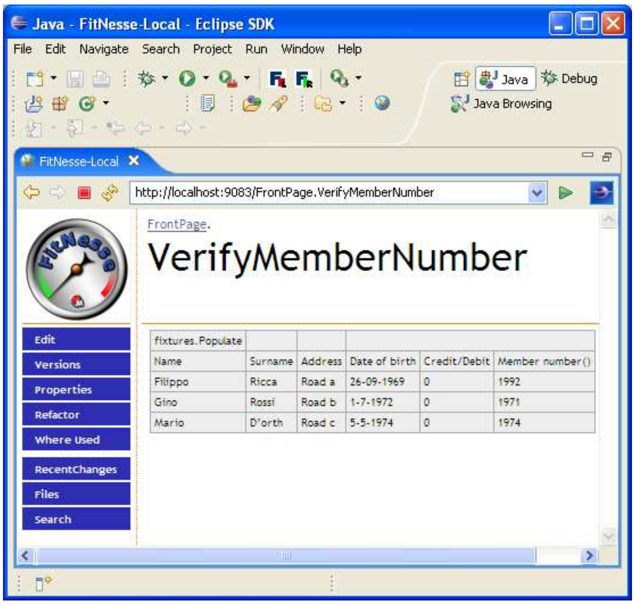
**Fit** (Framework for Integrated Test) is an open source framework used to express acceptance test cases and a tool for improving the communication between analysts and developers. Fit lets analysts write acceptance tests using simple HTML tables (**Fit tables**). The Fit tables serve as the input and expected output for the tests. Figure 1 shows an example of Column Fit tables, a particular kind of table (see [6] for the other types) where each row represents a test case. The first five columns are input values (*Name*, *Surname*, *Address*, *Date of birth* and *Credit/Debit*) and the last column represents the corresponding expected output value (*Member number()*).

Developers write code (**Fixtures**) to link the test cases with the System to verify. A component in the framework, the **Test Runner**, compares Fit table data with actual values obtained from the System. The test runner highlights the results with colors (green = correct, red = wrong).

The adoption of Fit tables is potentially a way to overcome the problems identified on change requirements. Fit tables represent an objectives way to specify change requirements, that cannot be interpreted in different ways by different people, as textual descriptions often are. Interpretation mismatches between analysts and developers should be hence highly reduced.

## 1.3 Research hypotheses

In this paper we describe an empirical study, devoted to assess whether Fit tables represent a better way to specify change requirements than natural language. We asked some students to answer questions about a software system, providing them the requirements, either in natural language or also in the form of Fit tables. The research questions that we are interested in answering are:



http://localhost:9083/FrontPage.VerifyMemberNumber

FrontPage.

VerifyMemberNumber

| fixtures.Populate | | | | | |
|---|---|---|---|---|---|
| Name | Surname | Address | Date of birth | Credit/Debit | Member number() |
| Filippo | Ricca | Road a | 26-09-1969 | 0 | 1992 |
| Gino | Rossi | Road b | 1-7-1972 | 0 | 1971 |
| Mario | D'orth | Road c | 5-5-1974 | 0 | 1974 |

**Figure 1: Example of Column Fit table. Fit table column's names without parenthesis represent input; parenthesis indicate output.**

**RQ1:** Does the presence of Fit tables[1] help programmers understand the requirements?

**RQ2:** Does the presence of Fit tables improve the *productivity* in the comprehension of textual requirements?

The context in which we investigate the above question has the following characteristics:

- System requirements have been written in detail.
- Some requirements are expressed only in textual form (-) while others include also Fit tables (+).

## 1.4 Related works

Although in the literature there are several papers [1, 8] and books [6] describing acceptance testing with Fit tables, only a few works report empirical studies about Fit.

The most related work is the paper by Melnik *et al.* [4]. It is a study focused on the use of Fit user acceptance tests for specifying functional requirements. It has been conducted at the University of Calgary and at the SAIT Institute (Southern Alberta Institute of Technology). In this experiment, the authors showed that the use of Fit tables and the possibility to execute them improve the comprehension of requirements.

The main differences from the present study are:

[1] Only Fit tables, not Fixtures. We are interested in understanding the possible benefits of adding Fit tables to requirements without considering additional benefits derived by test cases execution.

| | |
|---|---|
| **Goal** | Analyze the improvement produced by requirements augmented with Fit tables on comprehension tasks. |
| **Context** | Classroom exercise, requirements are provided on paper. |
| **Null hypothesis** | No effect on comprehension. |
| **Main factor** | Type of requirements used:<br>text only vs. text plus Fit tables. |
| **Other factors** | Specific requirements. |
| **Dependent variables** | Comprehension level and time. |

**Table 1: Overview of the experimental design.**

1. **Non-controlled vs. controlled experiment**. Melnik *et al.* [4]'s students worked on their own, off-line for two weeks (i.e., the experiment was unsupervised). In our case, students completed the tasks in a 2-hour laboratory without any possibility to exchange information.
2. **Control group**. In Melnik *et al.*'s study [4] all the students had the Fit tables to implement the change requirements. In our study each student received six requirements, three with Fit tables and three without them (*control group*).
3. **Working in teams vs. individuals**. In Melnik *et al.*'s study [4] students worked in team while in our experiment they worked alone.
4. **Implementation vs. questions**. In Melnik *et al.*'s study [4] students had to implement change requirements and the evaluation was done considering the number of test cases passed. In our case, students had to answer a comprehension question for each requirement. Differently from Melnik *et al.* [4] the evaluation was made by considering the number of correct and wrong answers.
5. **Executable test cases**. In Melnik *et al.*'s study [4] students had the Fit tables and the Fixtures with the possibility to execute them. In our case, students could only use the Fit tables to better grasp the requirements.
6. **Guidelines**. We designed the experiment following the guidelines by Wohlin *et al.* [10] and Juristo & Moreno [3].

## 1.5 Paper organization

The paper is organized as follows: Section 2 describes the design of the empirical study that we conducted. Results are presented in Section 3. Discussion, conclusions and future works are given respectively in Section 4 and Section 5.

# 2. EXPERIMENTAL DESIGN

We conceived and designed the experiment following the guidelines by Wohlin *et al.* [10] & Juristo and Moreno [3]. Table 1 summarizes the main elements of the experimentation.

## 2.1 Design

The *goal* of the study is to analyze the use Fit tables in requirements, with the *purpose* of evaluating their usefulness to improve the comprehension of requirements and effort. The *quality focus* is ensuring high comprehensibility and maintainability, while the *perspective* is both of *Researchers*, evaluating how effective are the Fit tables during the comprehension activities, and of *Project managers*, evaluating the possibility of adopting the Fit approach to augment application requirements. The *context* of the experiment consists of *objects* – a set of six requirements – and of *subjects*, students from a master course.

We adopt a very simple experiment design intended to fit a single 2-hours lab session. We have six objects and two treatments. The objects being the requirements for a single application, and the treatments being:

+ textual requirement enhanced with Fit tables

− textual requirement only

The subjects are given six requirements about a single software system. Then, they are asked to answer six questions (i.e., one per requirement).

The subjects are split into two groups (Red and Yellow), which are administered the combination of treatments shown in Table 2. Each group is provided some textual-only requirements (*e.g.*, Red group Q2, Q4 and Q6) and some textual requirements plus Fit tables (*e.g.*, Red group Q1, Q3 and Q5). The order of the requirements/questions is the same for the two groups, *e.g.* Q1 contains the same requirement (though expressed in different ways) and the same question for both groups.

One feature of this design is that the assignment received by the students in each group is of comparable difficulty and involves the same technologies and knowledge, thus it is ethically acceptable as a course assignment.

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 |
|---|---|---|---|---|---|---|
| Red | + | - | + | - | + | - |
| Yellow | - | + | - | + | - | + |

**Table 2: Experiment design**

## 2.2 Population

The subjects were 15 students from the course of Laboratory of Software Analysis, in their last year of the master degree (5th year or 2nd year laurea specialistica) in computer science at the University of Trento. The students belonged to various nationalities (Italian, Indian, Romanian, Pakistani Malaysian, etc.). The students had a good knowledge about programming and an average knowledge about software engineering topics (i.e., requirements, design and testing). Subjects have been trained in meaning and usage of Fit tables.

## 2.3 Material

The experiment was introduced as a lab assignment about Fit tables and Requirements.

Every subject received:

- Summary description of the application
- Six sheets (given one after the other) containing each:
  - a requirement with (+) or without (−) Fit tables.
  - a question about the requirement
  - specific slots to annotate the time spent to answer (start and stop time)
- A post experiment questionnaire

The post experiment questionnaire aimed at both gaining insights about the students' behavior during the experiment and finding justifications for the quantitative results. It included questions about the task and system complexity, the adequacy of the time allowed to complete the task and the perceived usefulness of the provided Fit tables.

## 2.4 Procedure

We followed a very well defined procedure during the experiment. Initially the subjects were given a short introduction to the experiment, then they were randomly assigned to either of the two groups and sit down in the classroom according to a typical checkerboard pattern. They could work on a single question per time, they were required to deliver the previous answer before they could work on the next one.

More precisely, the actual experiment went through the following steps:

1. We delivered a sheet containing the description of the system.
2. Subjects had 10 minutes to read the description of the system.
3. For each requirement in the sequence:
   (a) We delivered the requirement and question sheet number $I$
   (b) Subjects had to write their name and start time.
   (c) Subjects had to read and understand the requirement and the question.
   (d) Subjects had to deduce the answer using requirement statements and Fit tables (when provided).
   (e) Subjects had to write the answer.
   (f) When finished, subjects had to mark the stop time and call the educators.
4. Subjects were asked to compile the Post Experiment Questionnaire

## 2.5 Object

The object of the study is a set of requirements of a Library System that helps a library employee to manage the loan of books and tapes. Members can borrow, reserve or renew (i.e., extend a current loan) books and tapes.

The description of the system was originally presented by Callan in the book [2], pages 169-174. We have modified it slightly for our purposes: *"A library issues loan items to customers. Each customer must be known as a member and as such is issued a membership card that with a unique member number. Along with the membership number, other details on a customer must be kept such as a name, address, and date of birth. The library is made up of a number of subject sections. Each section is denoted by a classification mark. A loan item is uniquely identified by a number bar code. There are two types of loan items; language tapes and books. A language tape has a title, language (e.g., French), level (e.g., beginner) and authors. A book has a title and authors. An author has two fields: name and surname. A customer may borrow up to a maximum of 8 items. An item can be borrowed, reserved or renewed to extend a current loan. Each of these activity has a cost in Euro (borrowing a book costs 10 Euros while a tape only 5; if the member performs at least 3 operations - i.e., borrow, renew and/or reserve - in the same day, she/he receive a discount of 7 Euros). When an item is issued, the borrowing customer's membership number is scanned via a bar code reader or entered manually. If the membership is still valid and the number of items on loan less than 8, the procedure can proceed and the book bar code is read, either via the bar code reader or entered manually. If the item can be issued (e.g., it is not reserved) a receipt of the item is printed and then the item is issued. The library must support the facility for an item to be searched and for an update of items and members."*

Requirements have been deduced starting from this textual description. Some ambiguities and inconsistencies have been intendedly inserted (e.g., book search case-sensitive or case insensitive, partial or complete search, date format, etc.) in the requirements, considering real cases of ambiguities and inconsistencies that actually happened in the development of *EasyCoin*[2].

Below we show an example of requirement and related question used in the experiment (they correspond to R1 and Q1):

- **Requirement**. *The library employee can insert, delete or update a member. Each member has the following fields: unique member number, name, surname, address, date of birth and credit/debit. The member number is computed automatically by the software. This value is calculated summing day, month and year and subtracting to the result the number of letters of name and surname. If the value obtained is not unique then the software subtracts 1 to it.*
- **Question**. *Supposing that the employee has already inserted 3 members (Ricca Filippo, Road a, 26 September '69, 0; Rossi Gino, Road b, 1 July 1972, 0; D'orth Mario, Road c, 5/5/74, 0) what is the member number of D'amico Tino, 5 April '93 as computed by the System that we have to develop?*

Students had to understand the requirement and the question and they had to give the answer. Students provided with the Fit tables (Red group in this case, see Table 2)

[2] EasyCoin is a simple database program for coin collectors developed by the students of the University of Genova.

can use them (see Figure 1) to try to disambiguate the requirement. In this case the correct answer is 1991 and the possible ambiguities are the date format (2 or 4 digits) and whether considering the apostrophe as letter.

## 2.6 Metrics

For each requirement $i$ and subject $S$ we considered the following metrics:

1. $Time_i^S$, time required to read and understand the requirement and to answer the question;
2. $Correct_i^S$, whether the answer was correct or not;
3. $TimeRel_i^S$, relative time.

Relative time is a derived metric, it is computed normalizing the absolute time by the total time used by each subject.

$$TimeRel_i^S = \frac{Time_i^S}{\sum_{j=1}^{6} Time_j^S}$$

The rationale behind this metric is that more skilled subjects could complete the task in a significantly smaller time than less skilled one.

The post-experiment questionnaire (see Appendix A) contained nine questions. A first group of questions (Q1 through Q6) served the purpose of validating instrumentation source of the internal validity. They address the availability of sufficient time to complete the tasks, the clarity of the requirements, and the ability of subjects to understand them. Another two questions (Q7 and Q8) aim at measuring how much time is devoted to textual requirements and to Fit tables. Eventually the last question is devoted to measure the perceived usefulness of Fit tables. All the questions are on a five point ordinal scale and use (except Q7 and Q8) a Likert scale [7]. The post-experiment questionnaire is provided in Appendix A.

## 2.7 Detailed hypotheses

We can now define the detailed null hypotheses. We have one detailed hypothesis for the comprehension and two for the time/effort.

$HC_0$ there is no difference in the mean number of correct answers given to questions related to requirements defined with vs. without Fit tables.

$HTa_0$ there is no difference in the mean *absolute* time required to answer questions related to requirements defined with vs. without Fit tables.

$HTr_0$ there is no difference in the mean *relative* time required to answer questions related to requirements defined with vs. without Fit tables.

# 3. EXPERIMENTAL RESULTS

We had 15 subjects participating in the experiment, 8 in the Red group and 7 in the Yellow group.

The average number of correct response is 2.3, with a minimum of 1 and a maximum of 5. To analyze the number of correct answers, we built contingency tables and applied the Fisher's exact test, which is more accurate than $\chi^2$-test for small sample sizes.

The subjects used 8 minutes per requirement on average, with a minimum of 2 minutes and a maximum of 20. The actual distribution of times to complete an answer is shown in Figure 2, the distribution is not normal (Shapiro-Wilk p=0.0013). As far as time-related hypotheses are concerned, we use the Mann-Whitney test, both because of the small sample size and the non-normality of the data.

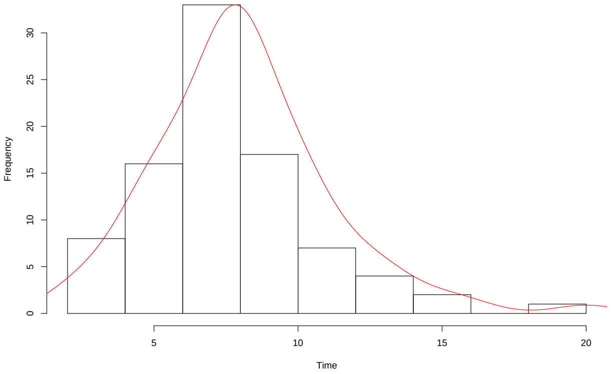


**Figure 2: Time to process a requirement.**

## 3.1 Correct answers

To test hypothesis $HC_0$ we consider the number of correct and wrong answers given by the subjects belonging to the two experimental groups. For each requirement we can build a $2 \times 2$ contingency matrix and apply Fisher's test to reject the null hypothesis. In Table 3, we present the number of correct answers. Numbers on a green (grey in B/W printing) background represent requirements augmented with Fit tables.

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 |
|---|---|---|---|---|---|---|
| Red | 6 | 3 | 7 | 1 | 3 | 0 |
| Yellow | 0 | 2 | 2 | 6 | 2 | 3 |
| *p*-value | 0.0069 | 1 | 0.04 | 0.01 | 1 | 0.076 |

**Table 3: Number of correct answers**

In addition we can also compare the two treatments in terms of overall correct answers, i.e., we consider the answer to different questions as independent measures. We obtain the contingency Table 4, the Fisher's test tells us that such a configuration exhibits a correlation between the treatment and the result with a significance level of $7.5 \cdot 10^{-5}$.

| | Wrong | Correct |
|---|---|---|
| Fit (+) | 18 | 27 |
| Text (−) | 37 | 8 |

**Table 4: Contingency table for correct answers.**

## 3.2 Time to answer

To address $HTa_0$ we consider the mean time required to process each requirement and answer the relative question. To test the hypothesis we apply the Mann-Whitney test. In Table 5 we present the mean times and the p-value of the statistical tests. The cells with the colored background represent requirements augmented with Fit tables.

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 |
|---|---|---|---|---|---|---|
| Red | 7.63 | 9.00 | 7.25 | 8.38 | 7.38 | 6.00 |
| Yellow | 9.83 | 10.86 | 6.00 | 9.57 | 7.86 | 8.00 |
| *p*-value | *0.007* | 0.41 | 0.16 | 0.68 | 1 | 0.19 |

**Table 5: Mean absolute time to answer.**

We follow the same procedure as for the previous hypothesis. We test the difference of the mean time required to answer *any* question. The average time required to answer a question relative to a requirement with Fit tables is 8'23", while it is 7'48" when Fit tables are not present. According to the results of the Mann-Whitney test ($p$-value=0.45), such a difference is not statistically significant.

The analysis procedure to test hypothesis $HTr_0$ is similar, the only difference being that here we deal with normalized times expressed as percentages of the total time required to complete all the tasks. Table 6 presents the mean relative times with the results of the Mann-Whitney tests. Considering the overall mean relative time, we have 17.7% for the requirements augmented with Fit tables and 16.4% for the requirements with text only. Such a difference is not significant ($p$-value=0.4).

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 |
|---|---|---|---|---|---|---|
| Red | 18% | 20% | 15% | 18% | 16% | 13% |
| Yellow | 20% | 22% | 12% | 19% | 16% | 17% |
| *p*-value | 0.24 | 0.31 | 0.091 | 0.60 | 1 | 0.17 |

**Table 6: Mean relative time to answer.**

## 3.3 Post experiment questionnaire

We coded the results from the first six questions of the questionnaire (see appendix A) on a Likert scale as follows: 1=strongly agree; 2=agree; 3=not certain; 4=disagree; 5=strongly disagree. The questions number seven and eight were coded according to the following schema: 1=<20%; 2=≥20% and <40%; 3=≥40% and <60%; 4=≥60% and <80%; 5=≥80%. The last question was coded in this way: 1=very much; 2=enough; 3=undecided; 4=little; 5=definitely not.

The data collected though the post-experiment questionnaire are summarized in Table 7.

The answers to questions Q1 through Q6 confirm that the subjects were able to understand the material provided to them within the time allowed by the experiment.

From the following two questions (Q7 and Q8) we find out that roughly 50% of the time was devoted to read the textual part of the requirements on the average. The Fit tables involved 30% of the time when present.

We can observe (Q9) that the subjects deemed the fit tables useful to understand better the requirements (median = "very much" and mean between "enough" and "very much"). The same is true in Melnik *et al.* [4].

## 3.4 Threats to Validity

We discuss the threats to the validity of the study according to the common partitioning into four categories: *internal*, *construct*, *conclusion* and *external* validity threats.

*Internal validity* threats are mainly due to the system we used as experimental object. While the system itself represents a real world application, requirements and the related questions are forcefully simple, thus they may be deemed as realistic but not real. On the other hand their size and complexity were designed to be proportional to the time available for experiment (a single 2 hour lab session).

*Construct validity* threats that may be present in this experiment, were addressed by using a fairly simple and standard design. Additionally, to avoid social threats due to evaluation apprehension, students were not evaluated on their performance in the Lab. Finally, subject were not aware of the experimental hypotheses.

About *conclusion validity*, proper tests were performed to statistically reject the null hypothesis. The small sample size (15 subjects) may limit the capability of statistical tests to reveal any effect; for contingency tables we used the Fisher's exact test, which is particularly suitable for such a context; while for the other analyses we used non-parametric tests.

Last, but not least, *external validity* threats are always present when experiments with students are conducted. Our results may be generalized to junior developers, but to draw any conclusions about more experienced developers we will need a controlled experiment with professionals. In any case, this is just a first piece of falsifiable knowledge that further studies with universities and industries could confirm or contradict.

# 4. DISCUSSION

We can reject $HC_0$ ($p - value = 7.517e - 05$) but neither $HTa_0$ (the direction is opposite) nor $HTr_0$ (the difference is not significant). The alternative hypothesis stating that "*fit tables are useful to understand better the requirements*" is supported by the direction, which is in favor of textual requirements plus Fit tables. It is also corroborated by the answers to the Question 9 of the post experiment questionnaire (median = "very much").

Students with Fit tables employ more time to answer the questions, even if the differences (absolute and relative time) are not significant. Probably this happens because they have to understand not only the textual requirements but also the tables. In the case of Fit tables, time could increase also because students have to verify the hypotheses deduced from the textual requirements directly on the examples (i.e., Fit tables). We think that only further empirical studies with more students could help us interpret the data obtained on time to answer and help us answer $HTa_0$ and $HTr_0$.

Some general remarks:

|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| mean | 1.33 | 1.93 | 2.00 | 2.13 | 2.20 | 1.60 | 3.40 | 2.27 | 1.53 |
| median | 1 | 2 | 2 | 2 | 2 | 1 | 3 | 2 | 1 |
| std dev | 0.49 | 0.80 | 1.00 | 0.83 | 0.68 | 0.83 | 0.91 | 0.88 | 0.83 |

**Table 7: Post-questionnaire data. Questions 7 and 8 are on time spent on Requirements vs. Fit Tables as perceived by subjects.**

1. Requirements were more ambiguous than expected (e.g., date format for the foreigners, unclear phrases, etc.).

2. Questions were more difficult and ambiguous than expected (only 32 correct answers out of 90 total, corresponding to only to 35.56% of correct answers).

3. In the explanation of the exercise we did not refresh the students about the syntax and semantics of Fit tables (i.e., action, column and row Fit tables [6]). We explained them one week before the experiment. With the initial questions, some students had difficulties to understand the Fit tables. Then, they checked syntax and semantics of Fit tables on the slides and the experiment could proceed normally.

## 5. CONCLUSION AND FUTURE WORKS

In this paper we have presented an experiment conducted with students in their last year of the master degree to assess whether availability of Fit tables affects the level of understanding and the productivity in understanding the requirements. Our preliminary experimental results agree well with those obtained by Melnik *et al.* [4]: they indicate that Fit tables improve requirement understanding (the difference is significant), but tend to involve additional effort (only in absolute time).

As it always happens with empirical studies, replication is the only way to corroborate our findings and try to resolve doubts. In particular we are interested in understanding better the relation between time and Fit tables and between absolute time and relative time. This to better answer RQ2.

It would be interesting to consider alternative experimental settings in several respects. In particular, we are interested in repeating the experiment with real requirements and with graduated students and professionals. It would be extremely important to understand how these different subpopulations of programmers make use of the Fit tables. This kind of studies is part of the agenda of our future work.

## 6. REFERENCES

[1] J. Aarniala. Acceptance testing. In *whitepaper. www.cs.helsinki.fi/u/jaarnial/jaarnial-testing.pdf*, October 30 2006.

[2] R. Callan. *Building Object-Oriented Systems: An Introduction from Concepts to Implementation in C++.* WIT Press (UK); BkDisk edition, 1994.

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

[4] G. Melnik, K. Read, and F. Maurer. Suitability of fit user acceptance tests for specifying functional requirements: Developer perspective. In *Extreme programming and agile methods - XP/Agile Universe 2004*, pages 60–72, August 2004.

[5] B. Meyer. On formalism in specification. *IEEE Software*, January 1985.

[6] R. Mugridge and W. Cunningham. *Fit for Developing Software: Framework for Integrated Tests.* Prentice Hall, 2005.

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

[8] K. Read, G. Melnik, and F. Maurer. Examining usage patters of the fit acceptance testing framework. In *Proc. 6th International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP2005)*, pages Lecture Notes in Computer Science, Vol. 3556, Springer Verlag: 127–136 2005, June 18-23 2005.

[9] F. Ricca, M. D. Penta, M. Torchiano, P. Tonella, and M. Ceccato. The role of experience and ability in comprehension tasks supported by uml stereotypes. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 375–384. IEEE Computer Society, May 2007.

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

[11] R. Young. *Effective Requirements Practice.* Addison-Wesley, Boston, MA, 2001.

# APPENDIX

## A. POST EXPERIMENT QUESTIONNAIRE

1. I had enough time to perform the lab tasks.
   □ strongly agree □ agree □ not certain □ disagree □ strongly disagree
2. The objectives of the lab were perfectly clear to me.
   □ strongly agree □ agree □ not certain □ disagree □ strongly disagree
3. The description of the System was clear.
   □ strongly agree □ agree □ not certain □ disagree □ strongly disagree
4. The questions asked were clear to me.
   □ strongly agree □ agree □ not certain □ disagree □ strongly disagree
5. I experienced no difficulty in reading/understanding the Requirements
   □ strongly agree □ agree □ not certain □ disagree □ strongly disagree
6. I experienced no difficulty in reading/understanding the Fit Tables.
   □ strongly agree □ agree □ not certain □ disagree □ strongly disagree
7. How much time (in terms of percentage) did you spend looking at Requirements?
   □ $<$20% □ $\geq$ 20% and $<$ 40% □ $\geq$ 40% and $<$ 60% □ $\geq$ 60% and $<$ 80% □ $\geq$ 80%
8. How much time (in terms of percentage) did you spend looking at Fit Tables?
   □ $<$20% □ $\geq$ 20% and $<$ 40% □ $\geq$ 40% and $<$ 60% □ $\geq$ 60% and $<$ 80% □ $\geq$ 80%
9. Did you find Fit tables (when available) useful to clarify requirements?
   □ very much □ enough □ undecided □ little □ definitely not