# Time Pressure: A Controlled Experiment of Test Case Development and Requirements Review

Mika V. Mäntylä
Aalto University,
P.O. BOX 19210, FI-00076,
Aalto, Finland
mika.mantyla@aalto.fi

Kai Petersen
Blekinge Institute of Technology
37140 Karlskrona, Sweden
kai.petersen@bth.se

Timo O. A. Lehtinen
Aalto University,
P.O. BOX 19210, FI-00076,
Aalto, Finland
timo.o.lehtinen@aalto.fi

Casper Lassenius
Aalto University,
P.O. BOX 19210, FI-00076,
Aalto, Finland
casper.lassenius@aalto.fi

## ABSTRACT
Time pressure is prevalent in the software industry in which shorter and shorter deadlines and high customer demands lead to increasingly tight deadlines. However, the effects of time pressure have received little attention in software engineering research. We performed a controlled experiment on time pressure with 97 observations from 54 subjects. Using a two-by-two crossover design, our subjects performed requirements review and test case development tasks. We found statistically significant evidence that time pressure increases efficiency in test case development (high effect size Cohen's d=1.279) and in requirements review (medium effect size Cohen's d=0.650). However, we found no statistically significant evidence that time pressure would decrease effectiveness or cause adverse effects on motivation, frustration or perceived performance. We also investigated the role of knowledge but found no evidence of the mediating role of knowledge in time pressure as suggested by prior work, possibly due to our subjects. We conclude that applying moderate time pressure for limited periods could be used to increase efficiency in software engineering tasks that are well structured and straight forward.

## Categories and Subject Descriptors
D.2.9 [Management]: Software quality assurance (SQA)

## General Terms
Experimentation, Human Factors, Verification.

## Keywords
Time pressure, Review, Test case development, Experiment

## 1. INTRODUCTION
The use and effect of time pressure on software engineering tasks has received limited attention in software engineering research. While we do not know the exact prevalence of time pressure in the software industry, we know that most projects (60-80%) encounter overruns [1]. As there always is pressure to complete a project on time and overruns are frequent, we can assume that time pressure is present in many software development projects.

In software engineering literature, time pressure is typically associated with negative outcomes. Time pressure:

- discourages careful planning and corrupts an engineering standard of quality [2]
- makes developers take short-cuts [3]
- reduces the time on software engineering activities [4]
- is a demotivator for software process improvement [5]
- causes failure to learn from mistakes [6]
- causes lower test case quality [7]
- is a factor of burnout in software teams [8]

Current trends such as the use of agile development with rapid release cycles and globally distributed development can increase time pressure. For example, the move to the rapid release model in the Firefox project forced the hiring of extra testing resources and reduction of regression testing coverage [9]. Furthermore a study of software testing proposed that global software engineering (GSD) increases time pressure [10].

Currently, many studies of time pressure in software engineering only capture the perceived effects of time pressure through qualitative data [5, 6, 10] or authors' negative beliefs [2-4, 7]. In fact, the scarce studies with none perceived data display a more diverse picture of time pressure in software engineering. Nan and Harter [11] found that medium time pressure on software development caused a reduction in the effort used and cycle time. Our past work found that time pressure increased defect detection efficiency in software testing by 71% [12]. Similarly, Jørgensen and Sjøberg [13] found that time pressure decreased the effort used on software engineering tasks. Furthermore, literature outside software engineering contains several studies on time pressure showing both positive and negative effects [14-16].

Still, current work on time pressure in software engineering context is insufficient. The prior studies have suffered from 1) non-equal settings between the treatment and control groups [12], 2) a small sample size [13], and 3) covering only database query creation tasks [17]. Furthermore, studies at the company level [11] data are influenced by confounding factors such as changes in the project scope, and personnel.

In this paper, we present a controlled experiment of the effects of time pressure in a software engineering context. We investigated the effect of time pressure by conducting an experiment with student subjects applying test case driven inspection [18], where requirements are reviewed while simultaneously creating high-level test cases based on the requirements. We measured the effect of time pressure by the time used, the number of defects detected and the quality of the test cases created. Furthermore, we analyzed the perceptions of the subjects with respect to the time pressure.

The rest of the paper is structured as follows. Section 2 develops a set of hypotheses based on previous studies on time pressure on


*ICSE'14*, May 31 – June 7, 2014, Hyderabad, India

software engineering, as well as in other fields. Investigation of these hypotheses is the main goal of this paper:

- *H1: Time pressure decreases effectiveness*
- *H2: Time pressure increases efficiency*
- *H3: Knowledge mediates the effect of time pressure*
- *H4: Time pressure is perceived negatively*

Section 3 explains our experimental design, while Section 4 presents the results of the paper. Section 5 discusses the results and provides avenues for further work, and finally, Section 6 contains our conclusions.

# 2. RELATED WORK

## 2.1 Effectiveness and Efficiency (H1 and H2)

Prior work on time pressure on software engineering tasks using measured effort, effectiveness and efficiency is limited. Our initial study [12] indicated that time pressure decreased effectiveness (less defects found in total) but increased efficiency (more defects found per time-unit) in software testing tasks. Similarly, Topi et al. [17] found that shorter time available was associated with decreased correctness on database query development tasks. Surprisingly, the study found no evidence supporting increased efficiency (correctness / minute). Additionally, Jørgensen and Sjøberg [13] found in a small experiment that time pressure reduced the effort spent, but also increased the number of errors in programming tasks. Unfortunately, measurements on the efficiency were not provided. Nan and Harter found that time pressure reduced development effort and cycle time in a software consulting company [11].

When we extend our scope beyond the domain of software engineering and information systems research, we can find more published research. In the domain of accounting, McDaniel [16] performed an experiment with 179 professional staff auditors and showed that time pressure and time restriction increased efficiency, but decreased the effectiveness of individual auditors matching our findings in [12]. In economic decision-making, Kocher and Sutter [19] found that time pressure reduced decision quality (reduced effectiveness). However, time pressure and time-dependent payoffs led to increased speed in decision making without reducing the accuracy of the decisions (increased efficiency) [19]. The authors think that "opportunity to gain considerably higher payoffs seems to trigger higher concentration or effort levels" and this helps to maintain high accuracy with increased speed in decision-making. Additionally, study of chess moves showed that games played under time pressure had lower move quality (decreased effectiveness) [20].

Studies [12, 13, 16, 20] suggest that time pressure decreases effectiveness, giving us Hypothesis 1 and the corresponding null hypothesis.

- H1: *Time pressure decreases effectiveness*
- $H1_0$: *Time pressure has no effect on effectiveness*

Studies [11-13, 16] suggest that time pressure increases efficiency, leading to Hypothesis 2.

- H2: *Time pressure increases efficiency*
- $H2_0$: *Time pressure has no effect on efficiency*

## 2.2 The Mediating Effect of Knowledge (H3)

High knowledge and skill is typically associated with automated and effortless mental processes [21]., that probably explains why knowledge mediates the effect of time pressure. There exists several examples of this outside software engineering literature.

In applied psychology, Beilock et al. [15] studied golfing under instructions that either (a) highlighted accuracy with taking as much time as needed or (b) instructed to perform as fast as possible while still being accurate. They found that novices produced typical speed-accuracy tradeoffs, i.e., playing faster decreased accuracy. However, experts performed better when playing faster. Furhermore, a study in accounting domain showed that accountants with high knowledge performed better under time pressure while low knowledge accountants performed worse [22]. Similarly, a study on chess players found that under time pressure, the quality of chess moves was reduced less for chess masters than for weaker players [20]. Thus, we get hypothesis 3:

- H3: *Knowledge mediates the effect of time pressure*
- $H3_0$: *Knowledge does not mediate the effect of time pressure*

## 2.3 Perceptions of Time Pressure (H4)

Empirical studies of perceptions of time pressure often mention time pressure as a negative factor. For example practitioners mention it as impediment to software quality [23], as a demotivator for software process improvement [5], as a factor of burnout [8], and as a decreaser of job satisfaction [24]. Thus, hypothesis 4 is as follows

- H4: *Time pressure is perceived negatively*
- $H4_0$: *Time pressure is not perceived negatively*

## 2.4 Summary of Literature

Table 1 lists the empirical evidence on time pressure in software engineering. The works that are based on the authors' opinions rather than empirical evidence are not included in the table.

**Table 1 Time pressure studies in software engineering.**

| Study, type and task | Outcome variables | Main result |
|---|---|---|
| [10] 2013, Industrial case study, Software testing, global software development | No measured outcomes (qualitative study) | Time pressure was perceived good and bad. Test teams experienced more time pressure than other teams. GSD alleviates the negatives of time pressure. |
| [12] 2013, Experiment, Software testing | Number of defects detected | Time pressure increased efficiency 71%. |
| [11] 2009, Industrial Case Study, Software projects | Cycle time and effort | Medium time pressure produced the highest productivity. |
| [17] 2005, Experiment, Database query development | Effort, correctness | Time pressure had no effect on effort or correctness. |
| [5], 2003, Industry interview study | None - Qualitative study) | Time pressure is a number one demotivator for process improvement. |
| [13] 2001, Experiment, Programming | Correctness, effort | Time pressure decreases effort used and correctness. (small sample size (10)) |
| [23] 1998, Industry interview study | None - Qualitative study | Lack of time was seen as a significant impediment to software quality. |
| [8] 1994, Industry survey | Factors of Burnout | Perceived pressure at work (not just time pressure) was a burnout factors in software development teams |

Table 2 shows examples of time pressure studies in other fields. The findings of these studies help in understanding human behavior under time pressure and are used later in the discussion section of this paper. The results can be summarized under categories of positive, negative and explanatory results.

- Positives: increased efficiency, less reliance on irrelevant information
- Negatives: decreases effectiveness, decreases job satisfaction, poorer results in tasks where interaction between humans is required
- Explanatory: knowledge and burnout rate mediates the impact of time pressure, time pressure leads to less risky and less cognitively demanding behavior

## 2.5 Prior Experiments on Reviews and Test case Development

Numerous software engineering experiments [25] have been conducted with many focusing on software reviews or inspections, e.g. [26-30]. However, according to our knowledge, prior experiments on software reviews or test case development have not assessed the impact of time pressure. Table 1 shows that time pressure has been the subject of experimental study only for programming [13], manual software testing [12] and database query development tasks [17]. Thus, it is important to study time pressure for other tasks as well.

We wanted to study time pressure in a context where subjects had an additional task on top of the review task. We were motivated by Fogelström and Gorschek [18], who studied test-driven inspection where subjects performed two tasks: inspection of requirements and simultaneous development of high-level test cases. They showed that the additional task of test case development did not impact the number of defects found in comparison to a situation in which only the inspection was performed. A hypothesis from [18] suggests that the additional task of test case development engages and forces the subjects to do a deeper review of the requirements. Thus, the additional task of test case development can be performed as "cost-free". A similar two-task setup has also been used by Klein et al [31], where subjects were required to perform pension calculation tasks while searching for defects in the pension data. In both prior works of two-task experiments, the tasks involved a single object of investigation, i.e. requirements [18] or pension data [31].

Furthermore, one of the tasks was defect detection while the other was something that forces the subject to process the object of investigation, i.e. develop high-level test cases [18] or perform pension calculations [31].

The two-task context has obvious benefits but also drawbacks. It allows investigating two tasks with a single experiment and, thus, can potentially allow quicker progress in understanding time pressure. The drawback is that we do not know how well the results can be compared with situation where only one task is performed.

**Table 2 Examples of studied of time pressure in other fields**

| | Study | Results |
|---|---|---|
| Effectiveness, quality, efficiency | [16] Experiment Accounting | Time pressure increased efficiency but decreased effectiveness. |
| | [19], Experiment, Economical decision making | Time pressure decreased decision quality. However, time pressure with time-dependent payoffs lead to faster decision without changing the decision quality. |
| | [14], Experiment Accounting audit | Time pressure reduced the adversary effects of using irrelevant information in accounting tasks. |
| | [32], Experiment Selecting a handball passing options | The first option chosen was superior in comparison to the option chosen after further consideration. |
| | [33], Employee survey and industrial data | Time pressure decrease patient safety on high burnout nurses only. |
| Role of knowledge | [15] Experiment Golfing / Putting | For novices, time pressure decreased accuracy. For experts time pressure increased accuracy. |
| | [20] Analysis of chess moves under different time-constraints | Grand masters' move quality decreased less under time pressure in comparison with lower level players. |
| | [22], Experiment Selecting correct keywords in accounting context | Time pressure improved performance in high knowledge, but decreased performance in low knowledge group. |
| | [34], Experiment Math problem solving accuracy | Time pressure leads to using less cognitively demanding problem solving strategies. |
| Interaction | [35], Industrial case study, buyer supplier relation | Reduces knowledge sharing in certain types buyer supplier relationships while not in others. |
| | [36], Experiment, negations results and stereotypes of participants | Time pressure results poorer agreements in negotiations and subject use more stereotypes of others. |
| Choice | [37], Experiment, Gambling task | Time pressure leads to less risky choice. |
| Job Satisfaction | [24], Employee survey | Time pressure decreased satisfaction at seven out of ten satisfaction types. |

# 3. RESEARCH METHODOLOGY

## 3.1 Experimental Design

We utilized a two-by-two crossover design [38], using two sessions and two groups of subjects, see Table 3. Part of the experiment material is also available online [39]. The treatment group was forced to work under time pressure, later referred to as the time pressure group (TP), while the control group was not working under time pressure. We refer to the control group as the non-time pressure group (NTP). The difference between sessions is the requirements specification that was used. In the first session, we used a requirements specification for an automatic teller machine (ATM), and in the second session we used a requirements specification for an online web-shop (OWS).

The experiment had two tasks, which were performed on a single object, i.e. the requirements specification of an ATM or an OWS. First, the subjects were asked to develop high-level test cases based on the requirements. Second, they were asked to review the specifications and record any defects they found in them. Both tasks had equal importance.

From prior work, we found three main approaches on how to create time pressure. The first option is to pre-test how much time is needed and then allocate insufficient time for the treatment group to create time pressure, as was done in [17]. This approach has the problem that different individuals work at a different pace. Thus, some individuals in the treatment group may be so fast that they do not experience time pressure at all, while the slowest individuals might experience so much time pressure that they give up. Therefore, time pressure created this way may have fluctuating effects on the individuals due to their individual speed. The second approach allows both the treatment and control group to use as much time as they like, but the treatment group is instructed to act as fast as possible, e.g., by giving them a low expectation on how much time they should use [13]. The problem with this approach is that the subjects may simply ignore the time pressure, as the completion of the task faster offers no benefits to the subjects. Finally, the third option is to have a linearly increasing incentive if the subjects complete the task faster, used in [19]. The third option was chosen for this experiment as it applies similar time pressure to all subjects regardless of the individual speed and it offers incentive for being faster.

**Table 3 Experimental design**

| | Group 1 | Group 2 |
|---|---|---|
| Session 1 (Requirements: ATM) | TP | NTP |
| Session 2 (Requirements: OWS) | NTP | TP |

## 3.2 Subjects and Incentives

Students on a Software Engineering course at Aalto University (Finland) had the possibility to voluntarily participate in the experiment. In total, we had 97 observations (49 TP and 48 NTP) from 54 students out of 168 who participated in the course. Students taking this course are typically half way through their studies on a five-year master's program that starts without a bachelor's degree. On average, the observations had completed 46% of the credits required for the master's degree with the standard deviation of 21%. In terms of industrial work experience in software development, the observations had higher fluctuation. A total of 37% of the observations had industrial work experience and the average, median, and standard deviation in years among the once having working experience were 4.3, 2.0, and 7.2 respectively. The prior experience in reviewing and testing that is presented Table 4. All data mentioned here was collected with our survey [39].

The students were awarded extra credit for the participation in the experiment. They got points for both participation and their performance in the experiment. As an ethical consideration we must mention that the points given based on this voluntary experiment based on this voluntary experiment performance were negligible for the students (3.6% of the total points).

In the time pressure condition the performance points where multiplied by a linear multiplier similar to one used in [19]. In our case, spending 45 minutes on the task gave a multiplier of one,

**Table 4 Prior experience in reviews and testing**

| Prior experience in (excluding pre-assignment) | Never | Done in courses | Done in industry |
|---|---|---|---|
| doing requirements review | 69% | 19% | 12% |
| developing test code | 24% | 62% | 14% |
| developing manual test cases | 46% | 41% | 12% |

using 5 minutes gave 1.8 and 85 minutes gave a multiplier of 0.2. Thus, starting from five minutes, the multiplier linearly declined by 0.02 for each minute spent in the experiment and the decline ended at 85 minutes. Figure 4 and Section 4.5 show the time used in the experiment and they indicate that our time pressure manipulation was successful.

## 3.3 Students as Subjects

The results of this paper have been obtained using student subjects, the use of which has been discussed both in software engineering [40-45] and in other fields [46]. Several papers indicate that student subjects do not have a significant difference compared to industrial subjects [40, 44-46] and the general conclusion appears to be that students can be used as long as they are trained and used to establish trends. In our experiment, all subjects were trained: they completed a pre-assignment based on instructions given to them. The pre-assignment had the same task as in the real experiment, but using a different requirements specification. Furthermore, this study focuses on finding trends rather than absolute effects of time pressure.

However, there are three possible sources of problems with student subjects. First, cultural differences might affect the student behavior. Luckily, many past empirical studies of student subject use have been conducted in Sweden [40, 44, 45] with a highly similar culture to Finland, as measured with Hofstede's cultural dimensions [47]. Second, Berander [45] presents data that the commitment of student subjects might bias the results. In our case the experiment was voluntary, thus the least motivated students are likely to be out of the subject sample. Furthermore, the subjects participating had a performance incentive as a small portion of the extra credit points were awarded based on the experiment performance. Finally, we surveyed the subjects' motivation in the experiment with a post-experiment survey question. Naturally, the motivation fluctuated between the subjects, but the correlation analysis revealed that there were no significant (alpha level 0.05) correlation between the reported motivation and the measured output: time used, the effectiveness or the efficiency. Thus, this indicates that our incentive manipulation mitigated the effect of motivation on the task performance. Third, Mortensen et al [46] points out that in the context of accounting tasks, students cannot be used as surrogates when the task is unstructured and complex. However, they also present data that when the task is structured and straight forward, using students as subjects is feasible. In this experiment, the task of creating test cases and reviewing requirements to find defects was well structured and straightforward.

## 3.4 Measures

The measures that we used to analyze the effectiveness are the number of requirements defects detected, and the score received from the test case development. Additionally, we calculated the efficiency, i.e. the requirements defects detected per hour and the test case score per hour. The test case score was based on the number of identified correct input and output variables, and created equivalence classes in the test cases.

Considering the requirements defects, most of them were known to the researchers in advance, as the same material had been used in a previous experiment [18]. Examples of requirement defects are: conflicting information in requirements or missing information, e.g. the rules of credit card withdrawal missing, what happens if ATM is out of money. All individual defects were assigned a unique ID. The subjects handed in their lists of defects at the end of both sessions. When a subject had found a defect that was not identified previously, a new ID was given for the defect. Later, all unique ID's were reviewed together by the first and third author to determine whether they represented a true defect or a false positive.

For the test case score, the correct input and output variables, and equivalence classes were pre-created by the authors. We further improved these measures based on the findings of the students that were not recognized by the authors, but were reasonable and valid. Using this type of process, where valid findings by the subjects improve the pre-created "correct solution" increases the validity of the results. The test case score is a sum of correct input and output variables and equivalence classes identified. For example in the ATM system the requirement for log-in could have a card as one input variable (1 point), equivalence classes for the card could be valid/invalid card (1 point) that could be split further to valid cards: bank, credit, bank+credit card and invalid cards: malfunctioning/not supported card (1 point).

The effort used was measured in minutes for each student individually. The starting time was marked at the beginning on a piece of paper and when the students returned their assignments, the experiment instructors marked down the individual end time. Thus, cheating on the start and end time was not possible.

Additionally, we surveyed the perceptions of subjects about the time pressure. After the experiment, each subject was given a survey form, see [39] for online version of the survey. The survey consisted of questions about their background, their motivation during the experiment, and their perceptions about the test case development and defect detection tasks. There were also questions about the working order that the students had followed and how well they could make a distinction about the two tasks. The perceptions of the tasks were collected using the NASA task load index, a well-known instrument for measuring task difficulty [48]. Importantly for this experiment, one of the six questions of the task load index enquired about the temporal demand of the task. Other questions assessed mental demand, physical demand, performance, effort, and frustration.

Furthermore, one of our hypotheses was related to the mediating role of knowledge on time pressure. Measuring knowledge in software engineering tasks is difficult and is currently lacking standardized tests, for an example of developing programming knowledge tests, see [49]. Thus, we collected several measures of the subjects' background knowledge with the survey form, see [39] for the survey, and Section 3.2 for the descriptive numbers of the measures. We used work experience and grade point average (GPA) of computer science courses (scale 1-5, avg/stdev: 3.7/0.59) in software development as knowledge measures for both review and test case development performance. GPA was collected from a course registration database instead of the survey to increase reliability. In addition, for review performance we used their prior experience in doing reviews. For test case development, we used prior experience in the development of manual test cases and experience in the development of automated tests. Thus, in total we used 5 measures of knowledge (3 for review performance and 4 for test case development performance, work experience and GPA used for both). To analyze the differences in knowledge for all cases, we split the subjects to two knowledge groups to high knowledge and low knowledge. We aimed for getting as equal sized groups as possible but due to distributions this was not always possible.

## 3.5 Statistical Analysis and Tests

Statistical tests were conducted using the R statistical computing program. We used the t-test to compare the outcomes between treatments when the assumption of normality was not rejected. Tests of normality were conducted using the Shapiro-Wilk test. When the data was not normally distributed, we used the Wilcoxon sum rank test (WRST). We report the effect size using Cohen's d [50] and 95% confidence intervals (CI) for it. For non-normal distributions, Cohen's d is computed using the R package Orddom [51], which converts the d values from Cliff's delta [52], which is a non-parametric effect size measure. We have opted to present all effect sizes as Cohen's d as interpreting two measures of effect size would be cumbersome for the reader. Positive Cohen's d values favor the TP and negative values the NTP group. A suggested way to interpret the effect sizes measured with Cohen's d are as follows: a value of 0.8 means a large effect, 0.5 means a medium effect, and 0.2 means a small effect [50]. Additionally, we conducted exploratory correlation analysis with Pearson and Spearman correlations to study relationships between variables. Finally for testing the interaction effect between knowledge and effectiveness we used ANOVA for normally distributed data and Poisson regression for non-normally distributed data. Analysis of distributions and the (non-central) Chi-Squared Distribution test of fit showed that using Poisson distribution was valid for the non-normally distributed data was valid.

# 4. RESULTS

## 4.1 Effectiveness (H1)

The first box-plot in Figure 1 illustrates that the defect count is slightly lower when working under time pressure (TP). The mean defect count for the TP group is 4.32 and 4.77 for NTP group. However, the difference is not statistically significant and the effect size favoring the NTP group is small. Regarding the test case score, second box-plot in Figure 1 illustrates that there are no differences between the groups. The mean test case score is slightly higher for the TP group than the NTP group—23.61 and 23.50 respectively. This small difference is not statistically significant and the effect size is almost non-existent. Table 5 summarizes these results.

**Table 5 Effectiveness between TP and NTP groups**

| | Defect count | Test case score |
|---|---|---|
| TP mean | 4.32 | 23.61 |
| NTP mean | 4.77 | 23.50 |
| p-value[1] | 0.342 | 0.922 |
| Cohens'd | -0.194 | 0.020 |
| 95% CI for d | -0.593 – 0.204 | -0.382 – 0.422 |
| Interpretation | Inconclusive | Inconclusive |

[1]Wilcoxon rank sum test and t-test for defect count and test case score respectively

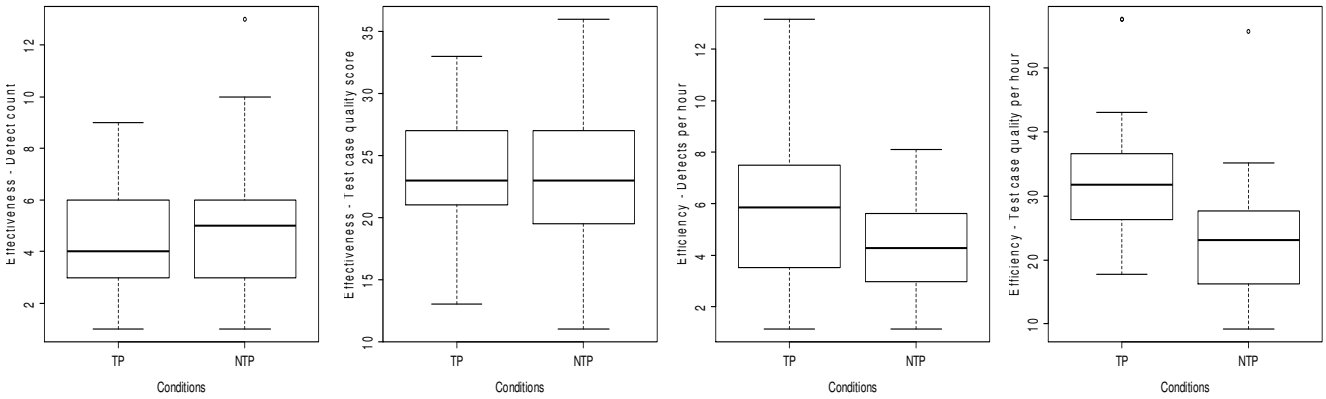


**Figure 1 Effectiveness and efficiency in time pressure (TP) and non-time pressure (NTP) conditions**

Exploratory correlation analysis revealed that higher individual time used correlated with higher effectiveness for both defect detection (r=0.43 level=0.001) and the test case score (r=0.21 level 0.05). Thus, spending more time resulted in a better outcome for both defect detection and test case development. However, the perceived time pressure, collected with NASA task load index, did not correlate with effectiveness.

Regarding defect detection, we have mixed results. Our results consolidate the hypothesis that time pressure reduces effectiveness, but only because less time is available. First, the TP groups found on average 0.45 fewer defects. However, the difference between the groups is too small to allow rejecting the null-hypothesis, see Table 5. Second, exploratory data-analysis showed a high and significant correlation between time used and the number of defects found. A non-supporting fact is that the perceived time pressure did not correlate with effectiveness. Thus, our interpretation is that time pressure reduces defect detection effectiveness, but only because less time is available, not because the pressure would have made our subjects less capable of finding defects.

Regarding the test case score, our results indicate that time pressure does not reduce effectiveness. The time pressure group was actually marginally better than the non-time pressure group. Additionally, perceived time pressure did not correlate with effectiveness.

## 4.2 Efficiency (H2)

The third and fourth box-plot in Figure 1 illustrates that efficiency is significantly higher when working under time pressure. The statistical tests are presented in Table 6. For the defect count, the mean number of defects per hour is 5.90 for the TP and 4.33 for the NTP group. The difference is statistically significant (p=0.002) and the effect size is medium (d=0.650). For the test case score, the mean scores per hour are 32.51 for the TP and 22.74 for the NTP groups. This difference is statistically significant (p=0.000) and has a high effect size (d=1.15).

Explorative correlation analysis shows that efficiency correlates with shorter working time. Regarding the test case scores, there is a high negative correlation between time used and efficiency (r=-0.72, level=0.001). For the defect detection, the negative correlation also exists, but it is smaller (r=-0.24, level=0.05). Additionally, perceived time pressure correlates with higher efficiency for the defect detection (r=0.28, level=0.01) and the test case development (r=0.29, level=0.01).

To summarize, our results indicate that time pressure increases efficiency in both defect detection and test case development. To support this hypothesis, we found three sources of statistically significant evidence: 1) the TP group had higher efficiency, 2) shorter time used correlates with higher efficiency and 3) perceived time pressure correlates with higher efficiency.

**Table 6 Efficiency between TP and NTP groups**

| | Defect count | Test case score |
|---|---|---|
| TP mean | 5.90 | 32.51 |
| NTP mean | 4.32 | 22.74 |
| p-value[1] | 0.002 | 0.000 |
| Cohens'd | 0.650 | 1.279 |
| 95% CI for d | 0.237 – 1.063 | 0.742 – 1.604 |
| Interpretation | Favors TP | Favors TP |

[1] Wilcoxon rank sum test and t-test for test case score and defect count respectively

## 4.3 Knowledge (H3)

Table 7 shows data from all interactions we studied. Defect count is studied of interaction effects with GPA, prior review and work experience. Test case score is studied of interaction effects with GPA, prior manual test case development, test automation and work experience. The data shows that in all except one case high-knowledge group benefits from time pressure where as low-knowledge suffers from it. The exception is that both high and low GPA students suffer from time pressure in defect detection. Furthermore, for test case development the differences are very small. Also for defect detection, the interaction effects of review and work experience were not statistically significant. Figure 2, a box-plot with mean lines, illustrates the interaction between prior review experience and time pressure with respect to defect count.

Thus, our data offers weak support to the prior theory of the mediating role of knowledge in the effect of time pressure. However, for test case score the differences are so small that they are in practice meaningless. For defect count, the differences are higher and they could be meaningful in practice. Finally, none of the knowledge measures are correlated with perceived time pressure. Thus, it seems that the individuals with different knowledge experience time pressure similarly.

**Table 7 Interaction between time pressure and knowledge. p-values are for interaction effect (calculated with Poisson regression for defect counts and ANOVA for Test case score)**

| Outcome variable | Knowledge (Experience) | Knowledge level | TP | NTP | p-value |
|---|---|---|---|---|---|
| Defect count | Work | High n=36 | **4.8** | 4.2 | 0.08 |
| | | Low n=61 | 4.1 | **5.1** | |
| | Review | High n=30 | **5.0** | 4.5 | 0.14 |
| | | Low n=67 | 4.0 | **4.9** | |
| | CS courses' GPA[1] | High=41 | 4.9 | **5.4** | NA[2] |
| | | Low=44 | 4.2 | **4.4** | |
| Test case score | Work | High n=36 | **24.3** | 23.2 | 0.58 |
| | | Low n=61 | 23.3 | **23.6** | |
| | Man. test case dev. | High n=52 | **23.5** | 23.2 | 0.90 |
| | | Low n=45 | 23.8 | **23.8** | |
| | Test automation | High n=74 | **24.2** | 23.5 | 0.32 |
| | | Low n=23 | 21.7 | **23.5** | |
| | CS courses' GPA[1] | High=41 | **26.1** | 25.5 | 0.81 |
| | | Low=44 | 22.3 | **22.3** | |

[1]Some subjects did not have any CS courses where grade other than pass/fail was given

[2]Raw data indicates no interaction, thus, no p-value is reported

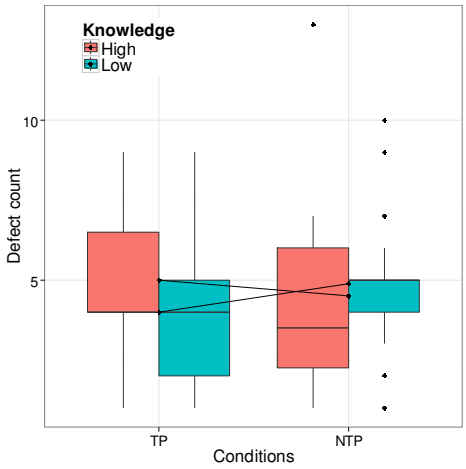


**Figure 2 Defect count and knowledge (review experience)**

## 4.4 Perceived Effects of Time Pressure (H4)

In the post experiment survey, we used the NASA task load index (NTLI) to measure the perceived effects of the tasks for the TP and NTP group. The NTLI has six questions that evaluate the task with respect to following criteria on a 20 point scale.

- Mental Demand - How mentally demanding was the task? – Scale: Very Low – Very High
- Physical Demand – How physically demanding was the task? – Scale: Very Low – Very High
- Temporal Demand – How hurried or rushed was the pace of the task? – Scale: Very High – Very Low
- Performance – How successful were you in accomplishing what you were asked to do? – Scale: Perfect – Failure
- Effort – How hard did you have to work to accomplish your level of performance? Very Low – Very High
- Frustration – How insecure, discouraged, irritated, stressed and annoyed were you? Very Low – Very High

We asked the questions for the defect detection and test case development tasks separately as we thought that the tasks would be perceived differently. However, we found no significant differences between the task load indexes for the two tasks. In fact, the perceptions of subjects about the tasks of defect detection and test case development had high and statistically significant correlations, r-values from 0.58 to 0.91 all with alpha level=0.001. This indicates that both the defect detection and test case development were perceived similarly by the subjects as measured by the NTLI.

Analysis revealed that only the questions about time pressure (Temporal demand) had a high and significant difference between the TP and NTP group. For the remaining questions, the differences were statistically insignificant, and the effect sizes small, see Table 8. Figure 3 show a comparison between the TP and NTP groups with respect to test case development and defect detection respectively. Regarding the frustration and performance of subjects, small adverse effects of time pressure in the defect detection task were found. However, the differences are statistically insignificant and the effect sizes small (d=0.26 and d=0.27). Similarly, under time pressure, some subjects felt that the tasks were physically more demanding. We think that this might be due to the need to increase the hand writing speed. However, these differences are also small and not statistically significant.

Furthermore, in the post-experiment survey, we asked the motivation of each subject during the experiment. The motivation was marginally higher in the TP than in the NTP group, but the difference was not statistically significant (WRST p=0.755, Cohen's d=0.06). Thus, we conclude that the adverse perceived effects of time pressure are small.

**Table 8 NASA task load index and tasks.**

| Task load index element | Task type | p-value (WRST) | Cohen's d (CI 95%) |
|---|---|---|---|
| Mental | TC | 0.652 | -0.06 (-0.42 – 0.37) |
| | RE | 0.444 | 0.11 (-0.25 – 0.55) |
| Physical | TC | 0.273 | 0.17 (-0.01 – 0.79) |
| | RE | 0.312 | 0.16 (-0.04 – 0.74) |
| Temporal | TC | 0.000*** | 1.44 (0.98 – 1.88) |
| | RE | 0.000*** | 1.71 (1.22 – 2.16) |
| Performance | TC | 0.800 | 0.04 (-0.34 – 0.46) |
| | RE | 0.096 | 0.27 (-0.04 – 0.76) |
| Effort | TC | 0.789 | 0.04 (-0.30 – 0.49) |
| | RE | 0.421 | 0.12 (-0.28 – 0.51) |
| Frustration | TC | 0.860 | 0.03 (-0.35 – 0.45) |
| | RE | 0.121 | 0.26 (-0.07 – 0.73) |

**Table 9 Motivation during the experiment**

| p-value (WRST) | Cohen's d (CI 95%) |
|---|---|
| 0.755 | 0.06 (-0.32 – 0.48) |

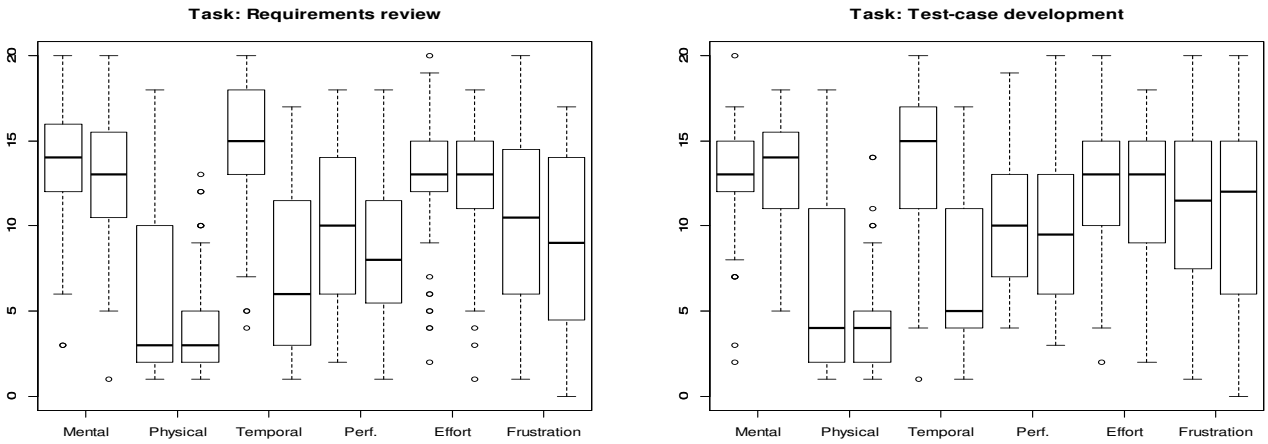


**Figure 3 NASA task load index. For each load index element TP is in left and NTP in right.**

## 4.5 Time Pressure and Time Used

Time pressure had a significant effect on the time used, as can be seen in Figure 4 (a box-plot with mean lines). The t-test indicates a high and statistically significant difference ($p<0.001$, Cohen's $d=1.42$) between the TP and NTP groups. Furthermore, for both the requirements review and test case development, the perceived time pressure measured with NASA task load index was significantly higher in the TP than it was in the NTP group ($d=1.71$ and $d=1.44$). To summarize, the time pressure had a real and significant impact on the subjects' time used and perceived time pressure.

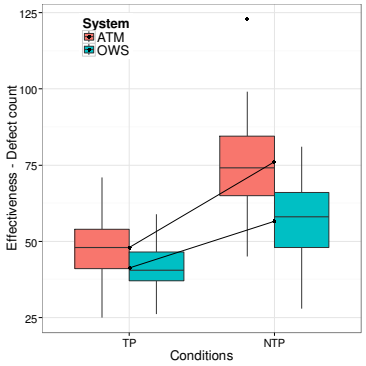


**Figure 4 Time used in TP and NTP conditions**

## 4.6 Effect of System and Learning

In this experiment, we cannot separate the effect of learning from the effect of system because ATM was always used first and OWS was always used second. Figure 5 (a box-plot with mean lines) shows the interaction between the system and the experimental conditions. The figure shows that defect detection of requirements review suffers from time pressure only in the ATM system. However, we cannot analyze whether this due to differences in the system or due to learning effects. Furthermore lower defect counts and test case scores originated from the OWS system.

We analyzed the interaction effect and variance for effectiveness, efficiency and time used with appropriate tests, see Section 3.5. Table 10 summarizes the statistical analysis. Poisson regression showed that defect count (effectiveness) was not significantly affected by the system type time pressure or interaction. ANOVA showed that there were no statistically significant effects in test case score affected by the system type, time pressure or interaction. For efficiency ANOVA showed that time pressure had significant effects to defect count and test case score while the system or interaction did not. For time used, time pressure, system and their interaction had significant effects

Thus, we conclude that the effect of system or learning did not have significant effects on the outcome results. However, the system or learning significantly affected the time used by the subjects.

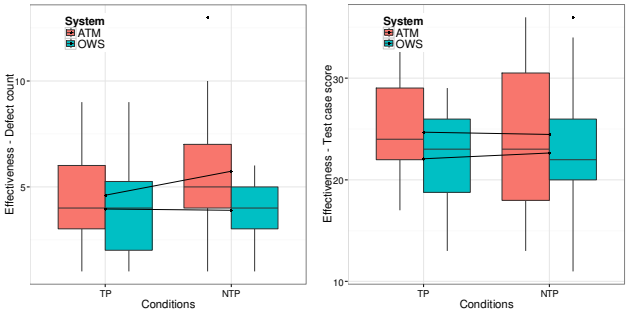


**Figure 5 Interaction between effectiveness and system**

**Table 10 p-values of variance and interaction effects.**

| | Effectiveness | | Efficiency | | Time used |
|---|---|---|---|---|---|
| Measure | Defect count | Test case score | Defect count | Test case score | |
| TP | 0.068 | 0.921 | *0.002* | *0.000* | *0.000* |
| System | 0.293 | 0.055 | 0.414 | 0.065 | *0.000* |
| Interaction | 0.215 | 0.711 | 0.811 | 0.229 | *0.016* |

# 5. DISCUSSION

Table 11 compares the results with respect to the performance variables. We can see that among the prior studies, there is agreement on the ability to save time using time pressure. In this

study, the fact of time saving is strongly supported, given the significant savings achieved.

**Table 11 Impact on Performance (H1-H3)**

| Effect and literature | This Experiment |
|---|---|
| (SE) **Increased efficiency**, captured as reduced cycle time and effort, increased productivity [11, 12] | (√) Significant increase in efficiency in defect detection and test case quality |
| (SE) **Negative effectiveness**, e.g. measured as poorer correctness (defects), correctness; [13, 17] | (•) Decreased effectiveness in defect count but not statistically significant<br>(!) No effect was observed in terms of quality of test cases developed |
| (OF) **Knowledge mediates the effect of time pressure,** e.g. low knowledge subjects suffer more from TP while high knowledge individuals may even benefit from TP [15, 20, 22] | (•) High knowledge individuals do better than low knowledge individuals in defect detection<br>(!) Test case development score is not affected by knowledge |
| (OF) **Less risky and cognitively less demanding behavior [34, 37]** | (?) Cognitive strategy or risk taking not investigated. Behavior with respect to willingness to take risk not investigated. |

**(√) = confirmed with statistically significant evidence, (•) = supporting evidence but not statistically significant (!) = contradict, (?) = Not investigated, SE = Software Engineering, OF = Other Fields**

Table 12 provides an overview of the outcomes with respect to the human factors. The factor of burn-out is not observable in this study as the observation is only conducted over a short period of time. Furthermore, there would be important ethical implications in trying to investigate burnout in a controlled setting. Hence, no evidence could be provided in this context.

**Table 12 Outcomes of Experiment in Relation to Literature for Human Factors (H4)**

| Effect | Experiment |
|---|---|
| Burnout [8] | (?) Not observable given that it would require long-term monitoring |
| Decreased job satisfaction, and motivation [5, 24, 53], | (•) Weak negative effects of time pressure to defect detection observed in the NASA task load index.<br>(!) No negative effects of time pressure to test case development task in the NASA task load index<br>(!) No difference in motivation |

## 5.1 Contextual factors and managerial implications

Our positive results of applying time pressure in the tasks of test case development and requirements review were related to a well-specified task for a limited period of time with moderate levels of time pressure. There are several important contextual factors that have an impact on the effect of time pressure. Here we position this study among those contextual factors and extend the discussion into the managerial implications.

**The amount of time pressure** affects whether time pressure improves performance or not. Time pressure has been shown to produce an inverted U-shaped diagram with respect to performance in software development, i.e., the optimal time pressure, that is neither too low nor too high, creates optimal performance. This is known as the Yerkes-Dodson law [54] and it has for example, been demonstrated that excessive financial rewards in time pressured tasks make people fail in the tasks they could otherwise complete [55], in other words people do choke under pressure. The time pressure in our study was related only to a small share of extra credit points given to the students, yet it still gave them an incentive to be fast. Thus, we think it is likely that our experimental conditions represented an optimal or a near optimal case of time pressure leading to high increases in efficiency. *Managerial implications:* Use moderate time pressure to increase efficiency. However, avoid excessive time pressure as it creates suboptimal performance. Both too large incentives and penalties should be avoided.

**Individuals' knowledge or skill** affects how people are affected by time pressure. Previous studies [15, 20, 22] indicated that knowledge has an important effect on how one reacts to and works under time pressure. Two studies conducted in completely different contexts (golf [15] and selecting keywords in accounting [22]) provide the same results, i.e. for experienced people, time pressure has a positive effect, while the effect is negative for people with low experience. In this study, the high knowledge individuals did better in defect detection (albeit not statistically significant), but not for test case development. Our results with prior work suggest that people with experience might benefit from time pressure, however, it should be avoided for people with low experience. *Managerial implications:* Most experienced and knowledgably individuals and teams are the best targets for time pressure.

**Task-type** affects whether time pressure is beneficial. A study of database query creation tasks did not find increased efficiency due to time pressure [17]. This conflicts with this work and [12]. We think the difference is due to tasks types. Steiner's taxonomy of tasks [56] claims that roughly speaking tasks can be either types of optimizing, which emphasizes the quality of the end result, or maximizing that emphasizes quantity. We think the database query creation tasks represent an optimizing task, i.e. you need only a single query but it has to be top quality / correct. Our tasks were more of the maximizing type, i.e. the goal was to find as many defects as possible and creating high quality test cases actually mean test cases that had the highest possible coverage, see Section 3.4. Thus, time pressure might be more suitable for tasks aiming at quantity rather than quality. Furthermore, our tasks were well structured and straight forward, see Section 3.3, but the same is true for [17]. *Managerial implications:* Tasks that require only the top quality and are complex, e.g. programming an encryption algorithm, are less suitable for time pressure. Tasks that require high quantity and are straight forward, e.g., finding as many defects as possible, are more suitable for time pressure.

There are also other dimensions in task type dimension than the quantity and quality. Findings outside software engineering indicate that interactive tasks suffer from time pressure [35, 36]. *Managerial implications:* Time pressure should not be applied to, for example, the inspection meeting, or other tasks with interaction tasks. However, the individual preparation for inspection meeting has potential of time savings through time pressure without significant impact on performance.

**Duration of time pressure** may affect whether time pressure is beneficial or not. Prior work has shown that time pressure has negative effects on job satisfaction and motivation [5, 24, 53]. Additionally, opinion based papers suggest that people working under time pressure might take shortcuts and corrupt the engineer-

ing standard of quality when executing processes [2, 3]. Such effects are well captured using the NASA task load index. Our investigation showed no significant effects of time pressure on perceived mental demand, physical demand, performance, effort, or frustration (including insecurity, being discouraged, irritated, and stressed). Furthermore, our subjects did not perceive effects on performance or taking shortcuts. Additionally, there was no difference in the motivation of subjects between the TP and NTP group. This discrepancy to prior work might be due to the duration of time pressure. It is possible that short duration of time pressure boosts efficiency without affecting motivation, but if the duration increases motivation starts to shrink. *Managerial implications:* Avoid long-term time pressure as it may lead to burnout and the loss of motivation. In large organizations, the savings of applying moderate time pressure for limited periods could be substantial when being applied to a high number of individuals, while it might be less relevant to small organizations.

## 5.2 Threats to Validity

We investigate the threats to validity by using the three viewpoints of internal, construct, and external validity [57]. Internal validity focuses on the causal relationship and statistical analysis used. It is affected by the violation of statistical test assumptions, and by low statistical power. We tested for the distribution of data when selecting statistical tests for hypotheses. Furthermore, effect sizes and their confidence intervals have been reported given that only checking for significance with respect to p-values is considered having limited value [58]. Also using a 2*2 design allowed to increase the sample size, and with that the statistical power.

Construct validity is concerned with whether the design and measures of the experiment actually capture what they intend to capture. In this study, these included the design of time pressure and the measurements on effectiveness, time used (efficiency), subject knowledge and perceptions of subjects. Regarding the design of time pressure, we think our design was unflawed as subjects used significantly less time and perceived more time pressure in the TP condition, see Section 4.5. Additionally, we used incentive based time pressure that should create more homogenous time pressure individuals as discussed in Section 3.1

Defect counts were used to measure effectiveness. However, this measure does not tell anything about the defect severity or impact. As the severity or impact of the defects was not analyzed, it is possible that there is difference in this respect between TP and NTP conditions. Furthermore, no analysis of the quality of the defect reports was done as it was enough that we were able to verify the defects. At the same, the test score was defined objectively, e.g. has a particular input variable been mention or has an equivalence class been formed (yes/no). Thus, the test case score measured the coverage of the developed test cases per subject only. However, it is possible that test cases developed in the TP group are less clearly written or have fewer words than those developed in NTP group. The uninvestigated quality of writing issues may pose a problem if one using the test cases or defect reports is a junior individual. In this paper, the primary evaluator (first author) had years of experience in teaching software testing and has plenty of experience in analyzing defect reports. Thus, in the future we should look at how interpretable the quality of reporting would be for less experienced individuals.

Cheating in the time used was impossible as researched registered starting time and end-time. Therefore, the risks for construct validity regarding the efficiency are similar than the risks of effectiveness.

We had several measures of knowledge that were analyzed. However, most of our knowledge measures were self-reported measures of experience, i.e. software development work experience and task experience specific measures. Furthermore, experience represents only a possibility to acquire knowledge [59]. We also measured GPA but it is unclear whether it might represent more discipline and general intelligence than knowledge about the tasks. Thus, we only had indirect measures of knowledge, which creates a construct validity threat.

The risks of construct validity related to the perceptions of subjects are minor. The NASA task load index is a well-established and tested instrument [48]. Additionally, the other questions of the survey that we used were easy to understand, see survey [39] for details.

Considering the external validity, there are two important risks that should be underlined. First is about using students as subjects, which was discussed in Section 3.3. We think that the conclusions of this study would not have changed even if we had used professional software engineers as subjects. The second risk is related to the transferability of the lab conditions to industrial settings. We cannot say anything about the effects of time pressure over a longer period of time. Furthermore, based on this study we cannot say what mitigation strategies time pressure might cause in real work situations. For example, in [9] we showed how the need to keep up with rapid releases caused the Firefox project to reduce regression testing scope, hire contractor resources while reducing the use of larger voluntary testing community. Naturally, with lab conditions such process changes that may be caused by time pressure cannot be captured. Thus, industrial studies about time pressure are needed.

## 5.3 Future Work

In future work, replications of this study are needed. It would be of great benefit to replicate this experiment with industry practitioners. Importantly, the industry practitioners would also act as a better source of high knowledge individuals than students.

Also, different types of tasks should be exposed and observed in relation to time pressure in controlled settings. The effect of time pressure on risk needs further work. Prior works indicate that risky behavior changes under time pressure [37]. This could have implications on software engineering studies that have studied the risks taken by product managers [60].

The overall number of studies focusing on time pressure is low, and hence there is no good understanding of the longitudinal effects on human factors, as well as outcome variables. Case studies and surveys in the area of software engineering would help to understand the effect of time pressure.

# 6. CONCLUSIONS

Time pressure has both positive and negative effects. Whether to use time pressure depends upon whether its positives outweigh the negatives. In the context of this experiment, we have demonstrated statistically significant benefits in terms of efficiency in test case development and defect detection with high and medium effect sizes. We found no statistically significant negative effects in terms of effectiveness or motivation, frustration or perceived performance. Weak non-significant adverse effects of time pressure were found with defect detection in requirements review. However, no adverse effects were found with test case development. Therefore, in this experiment, where moderate time pressure was applied for a limited period to well-structured and straight forward task, the positives outweigh the negatives.

## 7. REFERENCES


[1] Molokken,K. and Jorgensen,M., "A review of software surveys on software effort estimation," *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on,* 2003, pp. 223-230.

[2] Wirth,N., "A plea for lean software," *Computer,* vol. 28, no. 2, 1995, pp. 64-68.

[3] Austin,R.D., "The effects of time pressure on quality in software development: An agency model," *Information Systems Research,* vol. 12, no. 2, 2001, pp. 195-207.

[4] Costello,S.H., "Software engineering under deadline pressure," *ACM SIGSOFT Software Engineering Notes,* vol. 9, no. 5, 1984, pp. 15-19.

[5] Baddoo,N. and Hall,T., "De-motivators for software process improvement: an analysis of practitioners' views," *J.Syst.Software,* vol. 66, no. 1, 2003, pp. 23-33.

[6] Glass,R.L., "LOYAL OPPOSITION: Project Retrospectives, and Why They Never Happen," *IEEE Software,* vol. 19, no. 5, 2002, pp. 112-111.

[7] Juristo,N. and Vegas,S., "Using differences among replications of software engineering experiments to gain knowledge," *Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on,* 2009, pp. 356-366.

[8] Sonnentag,S., Brodbeck,F.C., Heinbokel,T. and Stolte,W., "Stressor-burnout relationship in software development teams," *J.Occup.Organ.Psychol.,* vol. 67, no. 4, 1994, pp. 327-341.

[9] Mäntylä,M.V., Khomh,F., Adams,B., Engström,E. and Petersen,K., "On Rapid Releases and Software Testing," *International Conference on Software Maintenance,* 2013, pp. 1-10.

[10] Shah,H., Harrold,M.J. and Sinha,S., "Global software testing under deadline pressure: Vendor-side experiences," *Information and Software Technology,* no. in Press,

[11] Nan,N. and Harter,D.E., "Impact of budget and schedule pressure on software development cycle time and effort," *Software Engineering, IEEE Transactions On,* vol. 35, no. 5, 2009, pp. 624-637.

[12] Mäntylä,M.V. and Itkonen,J., "More testers – The effect of crowd size and time restriction in software testing," *Information and Software Technology,* vol. 55, no. 6, 2013, pp. 986-1003.

[13] Jørgensen,M. and Sjøberg,D.I., "Impact of effort estimates on software project work," *Information and Software Technology,* vol. 43, no. 15, 2001, pp. 939-948.

[14] Glover,S.M., "The influence of time pressure and accountability on auditors' processing of nondiagnostic information," *Journal of Accounting Research,* vol. 35, no. 2, 1997, pp. 213-226.

[15] Beilock,S.L., Bertenthal,B.I., Hoerger,M. and Carr,T.H., "When does haste make waste? Speed-accuracy tradeoff, skill level, and the tools of the trade." *Journal of Experimental Psychology: Applied,* vol. 14, no. 4, 2008, pp. 340.

[16] McDaniel,L.S., "The effects of time pressure and audit program structure on audit performance," *Journal of Accounting Research,* vol. 28, no. 2, 1990, pp. 267-285.

[17] Topi,H., Valacich,J.S. and Hoffer,J.A., "The effects of task complexity and time availability limitations on human performance in database query tasks," *International Journal of Human-Computer Studies,* vol. 62, no. 3, 2005, pp. 349-379.

[18] Fogelström,N.D. and Gorschek,T., "Test-case Driven versus Checklist-based Inspections of Software Requirements–An Experimental Evaluation," *WER07-Workshop em Engenharia de Requisitos, Toronto, Canada,* 2007, pp. 116-126.

[19] Kocher,M.G. and Sutter,M., "Time is money—Time pressure, incentives, and the quality of decision-making," *Journal of Economic Behavior & Organization,* vol. 61, no. 3, 2006, pp. 375-392.

[20] Calderwood,R., Klein,G.A. and Crandall,B.W., "Time pressure, skill, and move quality in chess," *Am.J.Psychol.,* 1988, pp. 481-493.

[21] Sweller,J., "Cognitive load theory, learning difficulty, and instructional design," *Learning and Instruction,* vol. 4, no. 4, 1994, pp. 295-312.

[22] Spilker,B.C., "The effects of time pressure and knowledge on key word selection behavior in tax research," *Accounting Review,* 1995, pp. 49-70.

[23] Wilson,D.N. and Hall,T., "Perceptions of software quality: a pilot study," *Software Quality Journal,* vol. 7, no. 1, 1998, pp. 67-75.

[24] Linzer,M., Konrad,T.R., Douglas,J., McMurray,J.E., Pathman,D.E., Williams,E.S., Schwartz,M.D., Gerrity,M., Scheckler,W. and Bigby,J., "Managed care, time pressure, and physician job satisfaction: results from the physician worklife study," *Journal of General Internal Medicine,* vol. 15, no. 7, 2000, pp. 441-450.

[25] Sjøberg,D., Hannay,J., Hansen,O., Kampenes,V., Karahasanovic,A., Liborg,N.K. and Rekdal,A., "A survey of controlled experiments in software engineering," *IEEE Trans.Software Eng.,* vol. 31, no. 9, 2005, pp. 733-753.

[26] Fry,Z.P. and Weimer,W., "A human study of fault localization accuracy," *Software Maintenance (ICSM), 2010 IEEE International Conference on,* 2010, pp. 1-10.

[27] Kemerer,C.F. and Paulk,M.C., "The impact of design and code reviews on software quality: An empirical study based on PSP data," *IEEE Trans.Software Eng.,* vol. 35, no. 4, 2009, pp. 534-550.

[28] Walia,G.S., Carver,J.C. and Nagappan,N., "The effect of the number of inspectors on the defect estimates produced by capture-recapture models," *Proceedings of the 30th international conference on Software engineering,* 2008, pp. 331-340.

[29] Maldonado,J.C., Carver,J., Shull,F., Fabbri,S., Dória,E., Martimiano,L., Mendonça,M. and Basili,V., "Perspective-Based Reading: A Replicated Experiment Focused on Individual Reviewer Effectiveness," *Empirical Software Engineering,* vol. 11, no. 1, 2006, pp. 119-142.

[30] Biffl,S. and Halling,M., "Investigating the defect detection effectiveness and cost benefit of nominal inspection teams," *Software Engineering, IEEE Transactions On,* vol. 29, no. 5, 2003, pp. 385-397.

[31] Klein,B.D., Goodhue,D.L. and Davis,G.B., "Can humans detect errors in data? Impact of base rates, incentives, and goals," *MIS Quarterly,* 1997, pp. 169-194.

[32] Johnson,J.G. and Raab,M., "Take the first: Option-generation and resulting choices," *Organ.Behav.Hum.Decis.Process.,* vol. 91, no. 2, 2003, pp. 215-229.

[33] Teng,C., Shyu,Y.L., Chiou,W., Fan,H. and Lam,S.M., "Interactive effects of nurse-experienced time pressure and burnout on patient safety: a cross-sectional survey," *Int.J.Nurs.Stud.,* vol. 47, no. 11, 2010, pp. 1442-1450.

[34] Beilock,S.L. and DeCaro,M.S., "From poor performance to success under stress: working memory, strategy selection, and mathematical problem solving under pressure." *Journal of Experimental Psychology: Learning, Memory, and Cognition,* vol. 33, no. 6, 2007, pp. 983.

[35] Thomas,R.W., Fugate,B.S. and Koukova,N.T., "Coping with Time Pressure and Knowledge Sharing in Buyer–Supplier Relationships," *Journal of Supply Chain Management,* vol. 47, no. 3, 2011, pp. 22-42.

[36] De Dreu,C.K., "Time pressure and closing of the mind in negotiation," *Organ.Behav.Hum.Decis.Process.,* vol. 91, no. 2, 2003, pp. 280-295.

[37] Ben Zur,H. and Breznitz,S.J., "The effect of time pressure on risky choice behavior," *Acta Psychol.,* vol. 47, no. 2, 2. 1981, pp. 89-104.

[38] Wohlin,C., Höst,M., Runeson,P., Ohlsson,M.C., Regnell,B. and Wesslén,A., *Experimentation in software engineering: an introduction,* Kluwer Academic Pub, 2000.

[39] M. V. Mäntylä. , "Experiment materials," 2013, Accessed 2013 http://users.tkk.fi/~mmantyla/TP/

[40] Höst,M., Regnell,B. and Wohlin,C., "Using students as subjects—a comparative study of students and professionals in lead-time impact assessment," *Empirical Software Engineering,* vol. 5, no. 3, 2000, pp. 201-214.

[41] Tichy,W.F., "Hints for reviewing empirical work in software engineering," *Empirical Software Engineering,* vol. 5, no. 4, 2000, pp. 309-312.

[42] Carver,J., Jaccheri,L., Morasca,S. and Shull,F., "Issues in using students in empirical studies in software engineering education," *Software Metrics Symposium, 2003. Proceedings. Ninth International,* 2003, pp. 239-249.

[43] Runeson,P., "Using students as experiment subjects–an analysis on graduate and freshmen student data," *Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering.–Keele University, UK,* 2003, pp. 95-102.

[44] Svahnberg,M., Aurum,A. and Wohlin,C., "Using students as subjects-an empirical evaluation," *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement,* 2008, pp. 288-290.

[45] Berander,P., "Using students as subjects in requirements prioritization," *Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 International Symposium on,* 2004, pp. 167-176.

[46] Mortensen,T., Fisher,R. and Wines,G., "Students as surrogates for practicing accountants: Further evidence," *Accounting Forum,* 2012,

[47] Anon. , "Power Distance Index," 2009, Accessed 2013 9/7 http://www.clearlycultural.com/geert-hofstede-cultural-dimensions/power-distance-index/

[48] Hart,S.G. and Staveland,L.E., "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," *Human Mental Workload,* vol. 1, no. 3, 1988, pp. 139-183.

[49] Bergersen,G.R., Hannay,J.E., Sjoberg,D.I., Dyba,T. and Karahasanovic,A., "Inferring skill from tests of programming performance: Combining time and quality," *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on,* 2011, pp. 305-314.

[50] Cohen,J., *Statistical power analysis for the behavioral sciences,* Lawrence Erlbaum, 1988.

[51] J. J. Rogmann. , "orddom: Ordinal Dominance Statistics," 2013, Accessed 2013 9/9 http://cran.r-project.org/web/packages/orddom/

[52] Cliff,N., "Dominance statistics: Ordinal analyses to answer ordinal questions." *Psychol.Bull.,* vol. 114, no. 3, 1993, pp. 494.

[53] Beecham,S., Baddoo,N., Hall,T., Robinson,H. and Sharp,H., "Motivation in Software Engineering: A systematic literature review," *Information and Software Technology,* vol. 50, no. 9-10, 2007, pp. 860-878.

[54] R.M. Yerkes and J.D. Dodson, "The relation of strength of stimulus to rapidity of habit-formation," *Journal of Comparative Neurology and Psychology,* vol. 18, no. 5, 1908, pp. 459-482.

[55] Ariely,D., Gneezy,U., Loewenstein,G. and Mazar,N., "Large stakes and big mistakes," *Rev.Econ.Stud.,* vol. 76, no. 2, 2009, pp. 451-469.

[56] I.D. Steiner, *Group Process and Productivity,* New York, New York, USA: Academic Press, 1972.

[57] Campbell,D.T. and Stanley,J.C., *Experimental and quasi-experimental design for research,* Chicago, USA: Rand Mcnally College Publishing Company, 1966.

[58] Dybå,T., Kampenes,V.B. and Sjøberg,D.I., "A systematic review of statistical power in software engineering experiments," *Information and Software Technology,* vol. 48, no. 8, 2006, pp. 745-755.

[59] Bonner,S.E. and Lewis,B.L., "Determinants of auditor expertise," *Journal of Accounting Research,* vol. 28, 1990, pp. 1-20.

[60] Fogelström,N.D., Barney,S., Aurum,A. and Hederstierna,A., "When product managers gamble with requirements: Attitudes to value and risk," in *Requirements engineering: Foundation for software quality,* Springer, 2009, pp. 1-15.