UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Taavi Karelson

Lab Package: Scriptless GUI Testing with TESTAR

Bachelor's Thesis (9 ECTS)

Supervisors: Dietmar Pfahl
Faiz Ali Shah

Tartu 2023

Lab Package: Scriptless GUI Testing with TESTAR

Abstract:

The goal of this thesis is to update and modernize the lab material about automated GUI testing for the course Software Testing (LTAT.05.006) at the University of Tartu. The thesis gives an overview of changes to previous lab materials, introduces the produced materials for the lab, analyzes the feedback gathered from the students, and gives potential improvement ideas for the material. The lab was carried out in the 2023 spring semester.

Keywords:

TESTAR, software testing, lab package

CERCS: P170, Computer science, numerical analysis, systems, control

Praktikumimaterjal: Stsenaariumita graafilise kasutajaliidese testimine kasutades testimise tarkvara TESTAR

Lühikokkuvõte:

Käesoleva bakalaureusetöö eesmärk on loodud graafilise kasutajaliidese testimise praktikumimaterjali uuendamine ja täiendamine Tartu Ülikooli kursuse “Tarkvara testimine” (LTAT.05.006) jaoks. Antud töös kirjeldatakse ellu viidud muudatusi võrreldes eelneva materjaliga, töö käigus loodud materjale, analüüsitakse tudengite tagasisidet ja tehakse ettepanekuid materjalide paremaks muutmiseks. Loodud materjale kasutati 2023. aasta kevadsemestril.

Võtmesõnad:

TESTAR, tarkvara testimine, praktikumimaterjal

CERCS: P170, Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1. Introduction

Software testing is a process that evaluates the system under test (SUT) to identify failures and other issues which could impact its functionality. Out of the various types of software testing that is covered in the course Software Testing (LTAT.05.006) at the University of Tartu, this thesis concentrates on Graphical User Interface (GUI) testing, where the testing tool generates the scripts automatically.

GUI testing is a type of software testing that aims to ensure that GUI functionality and usability meet the requirements. This is usually done manually, but since manual testing is a long and tedious procedure, then it is also prone to human errors. To tackle this type of errors, automation tools have been created to save user interactions with the SUTs as scripts, which can then be replayed, therefore saving human time and effort. To minimize the time spent on maintaining the scripts that are prone to break if GUI layouts or API endpoints change, a new type of software testing called scriptless GUI testing has been developed.

The purpose of this thesis is to update and modernize the previously developed lab package "Automated GUI Testing ", which used the test automation tool TESTAR.

This thesis consists of six sections:

- Section 1 is the introduction of the thesis.
- Section 2 gives a brief background of GUI testing and the TESTAR tool.
- Section 3 provides an overview of the materials and tasks used in the lab.
- Section 4 describes the execution of the lab.
- Section 5 focuses on the feedback from the student.
- Section 6 is the summary of the thesis.

2. Background and existing materials

This section gives an overview of the Software Testing course, a brief background of GUI testing and its limitations, and introduces the tool used for this lab package.

2.1. Software Testing course

Software Testing (LTAT.05.006) is a 6 ETCS course at the University of Tartu that is taught in the spring semesters. This course is part of an optional IT-Systems Speciality Module (24 ETCS), out of which one of the objectives is: "Knows different testing techniques and is able to carry out testing planning and documentation" [1]. According to Student Information System, the Software Testing course objective is to "addresses the essential concepts of software quality control and testing and introduce various testing strategies and types of testing. It will also give an overview of different software defects, software defect management, and organizational aspects of software testing" [2].

In the 2022/2023 academic year, this course consists of 11 labs. The topics of those labs are the following [3]:

1. Debugging
2. Basic Black-Box Testing
3. Combinatorial Testing
4. Basic White-Box Testing
5. Random Testing
6. Automated Web-Application Testing
7. Scriptless GUI Testing with TESTAR
8. Security Testing
9. Mutation Testing
10. Static Code Analysis
11. Document Inspection and Defect Prediction

This thesis will focus on the 7th lab: Scriptless GUI Testing with TESTAR.

2.2. Automated Testing

Emil Börjesson and Robert Feldt [4] point out in their paper that due to the demand for faster time-to-market and higher-quality software, the need for a better solution than manual testing is needed. Often companies are challenged by their systems because those systems are GUI intensive and, due to that, complex and costly to test. This comes down to the need for continuous changes to the system, refactoring, etc. Therefore, companies tend to test only specific parts that have been changed; therefore, faults tend to slip through those tests, which affects the overall systems. Automated testing is seen as one solution to that problem because it allows one to run the tests faster and more often than a manual tester could.

2.3. Record and Replay

According to Börjesson E. [5], the most common GUI testing approach is called Record and Replay (R&R), also known as Capture and Replay. R&R uses two steps in its workflow. Firstly, user interactions with the SUT get recorded and saved as scripts. Secondly, the generated scripts can be replayed to rerun the tests automatically. Different tools record the user actions on different GUI abstraction levels, which all have various advantages and disadvantages. For example, using the GUI bitmap level, the system gathers information regarding the SUT coordinates from its bitmaps and records those coordinates of the GUI interactions in a script. This approach is sensitive to GUI layout reconfiguration but robust to API component changes. But this raises an issue that this kind of test would be hard to maintain.

2.4. TESTAR

To tackle the maintenance problem, TESTAR is an open-source tool that carries out automated testing without requiring scripts [6]. This means that no test cases need to be defined prior to test execution. Instead, all of the steps are generated randomly throughout the execution.

How TESTAR works [6]: After opening the SUT, it uses operating system accessibility APIs (e.g., UIA Automation for Windows) or other SUT-specific APIs to collect information regarding the visible widgets for the current state of the GUI. Out of that state, all enabled, unblocked, not blacklisted, or filtered actions get derived with possible actions like click, type into, drag, or slide. After that, an action is randomly selected and executed. This step is followed by the evaluation of new states to find failures. TESTAR uses oracles as mechanisms whether the test case passes or fails. Oracles are suspicious keywords that the user can define, and if those are encountered in any widget property, then the test case fails. On top of the oracles, TESTAR catches SUT crashing and freezing. In case of failure is detected, the sequence ends, and the sequence that leads to failure is saved. If no failure is caught, then the stop criteria specified by the user are checked. If stopping criteria is not met, then the flow repeats from scanning the current state of the current GUI. But if stopping criteria are met, the SUT is closed, and a test sequence is saved.

At the end of each sequence HTML report, a replayable TESTAR sequence, a log file, and screenshots, that are taken after each action, get saved. Those saved files allow users to identify if something suspicious was found.

2.5. Previous TESTAR Lab Package

In 2019, Prink Kert [7] created an "Automated GUI Testing" lab package for the Software Testing course. His work was based on TESTAR version 1.3. As of 11. October 2022, the most recent version of TESTAR is 2.5.3, and the "Scriptless GUI Testing with TESTAR" material is based on that. Two of the most significant improvements between those two versions are Record mode, which allows users to manually record sequences, and WebDriver support, which enables the testing of web applications. The last one is used in this work to improve the complexity of previous lab package homework.

As stated in Prink's thesis [7], a configured TESTAR should be provided to students. In the "Scriptless GUI Testing with TESTAR" lab materials, this was covered by setting up lab materials so that all the different steps of the TESTAR setup came in a logical order and had sufficient examples.

2.6. Scriptless vs. Scripted Testing

Axel Bons, Beatriz Marín, Pekka Aho, and Tanja E.J. Vos [8] did a case study to evaluate the complementary of automated GUI testing tools. As a scriptless testing tool, they used TESTAR, and as a scripted tool, they used Selenium with 11 scripts. The study was performed in an IT development company called E-Dynamics in the Netherlands. The software used in the study was Yoobi, a time-tracking web and mobile application. From an effectiveness perspective, as seen in Table 1, manual testing found four failures. Selenium also found four, but out of those, two were not caught manually. TESTAR found nine failures, two of which were caught manually and one by all three tests. Event coverage with Selenium was 3.06%, and with TESTAR, it was 8.37%. But it is reasonable to point out that Selenium found four failures with high severity, which was two more than TESTAR. From an efficiency perspective, manual testing took 32,5 hours, using Selenium took 27,5 hours, and using TESTAR took 34,5 hours. It was also pointed out that both tools took in total for 6 test subjects around 14 hours to learn. And the testing phase evaluation for Selenium took 1 hour in total, but for TESTAR, it took 5 hours in total.

Table 1. Effectiveness of TESTAR and Selenium [8].

Description	Severity	Manual	Selenium	TESTAR
Injected javascript error to test custom verdict.	None	X		X
Injected cfdump output in code to test custom verdict class detection.	None	X	X	X
Injected new label to test custom verdict, missing translation text detection.	None	X		X
Project edit/create does not work after API changes.	High	X	X	
Javascript error when switching the list detail view to only detail.	High			X
Missing label in supervisor menu.	Low			X
Two suspicious titles for flexmonter (js report plugin).	Low			X
Expense add error because of debug data in data-call.	High		X	
Failure when editing and creating a task due to the last release change.	High		X	
Missing translation in optional filter for activity in bulkmutation.	Low			X
Javascript error when opening the tour.	High			X
Never used missing translation in source code.	Low			X

Their conclusion was that scriptless and scripted approaches are complementary, and they improve the manual testing process. They also pointed out that Selenium was better at detecting process failures, and TESTAR was better at detecting visible failures and also achieved higher event coverage.

3. Lab Design

The upcoming chapter gives an overview of the structure of the lab "Scriptless GUI Testing with TESTAR". It provides a timeline for the lab, descriptions of materials used in the lab, information regarding lab and homework assignments, and grading.

3.1. Lab Schedule

Each lab in the course is expected to last for 8 academic hours (360 min): 2 hours (90 min) for the lab and 6 hours (270 min) for the homework task. The lab schedule is as follows:

- 25 minutes for installation of the VirtualBox and downloading and importing the virtual drive
- 20 minutes for introduction and initial setup of the TESTAR tool
- 40 minutes for a lab assignment
- 5 minutes for introduction to homework

The homework assignment can be solved alone or in pairs with students from the same lab group.

3.2. Lab Materials

This lab has two types of materials: lab material for students and lab material for teaching assistants (TAs). This sub-section gives an overview of those materials and of lab and homework tasks. Those files are provided as links in the appendixes.

3.2.1. Student Materials

Lab Instructions for Students is a guide for the whole lab. It explains the installation of VirtualBox, includes introduction to GUI testing, the introduction of the TESTAR tool, a step-by-step guide to the initial setup of TESTAR, lab and homework tasks with samples, grading information, helpful links and references, and finally, appendixes for lab and homework tasks.

Slides support TAs throughout the lab session. This includes the setup of VirtualBox, the introduction of the TESTAR tool, a step-by-step guide to the initial setup of TESTAR, grading information, and hints for homework.

Testar.zip is a compressed file of TESTAR version 2.5.3, the newest version of that tool when the lab materials were developed. This container also includes the lab SUT "Part1_calculator.jar".

Windows_10_testar.ova file for a virtual machine is an open virtualization format file. It has a Windows 10 virtual machine on it. The virtual machine has the TESTAR tool (same as the testar.zip) and has preinstalled: Java 8, Java JDK 16.0.2, Chromium version 110, and WebDriver for Google Chrome 110, which are required for running TESTAR and the SUTs. The virtual machine is used to provide students with a preset and stable system on which the tasks have been tested beforehand.

3.2.2. TA Material

Lab Instructions for TAs is a more comprehensive version of the Students Materials, including extra information regarding the lab session to support TAs if students have questions regarding the lab materials. It also provides extra hints and solutions for the lab and homework assignments.

3.3. Lab Session

In the lab, students have to first download the .OVA file and open the virtual machine, or if they are using the Windows operating system, download and unzip the Testar.zip file and download the required software. The in-lab assignment is meant to introduce the TESTAR tool and help with homework.

For that, the TAs use slides provided in the Student materials to show how to set up TESTAR so that they would be able to find the first failure in the system. Students are shown how to set filters, so TESTAR would know which buttons not to click. Then an oracle is set to capture the SUT state that can be seen in Figure 1. In the lab session, a faulty calculator application developed in Java 8 by Kert Prink for the “Automated GUI Testing” [7] lab package is used.

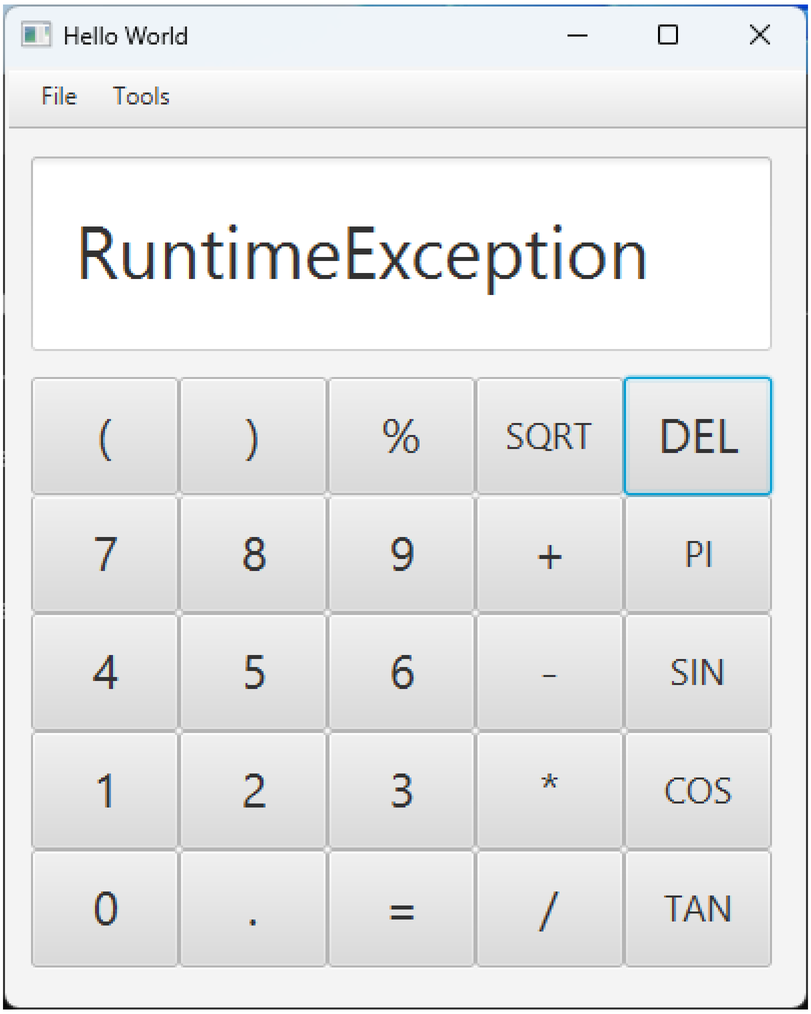

Figure 1. Lab session SUT failure message.

Even though the Lab Materials provide instructions on how to find this failure, students might not find it, as TESTAR selects actions randomly. Therefore, they have to use their freshly learned knowledge and provided requirements (in the appendixes of the Lab Instructions for Students) for that calculator to supplement their filters and oracles. While doing that, they have to fill in a failure report, which can be found in the appendixes. The lab session failure report is not graded; it aims to show how to report the homework assignment.

3.4. Homework Assignment

The SUT for the homework is a calculator web application created using the Vue.js framework. It has two views, as seen in Figures 2 and 3. This application was inspired by Kert Prinks' thesis [7] "Future Improvements" section, where he stated that a higher complexity homework task should be implemented. Using WebDriver protocol allows using additional settings to handle popups and console errors. Those settings in TESTAR are not available for Java application testing.

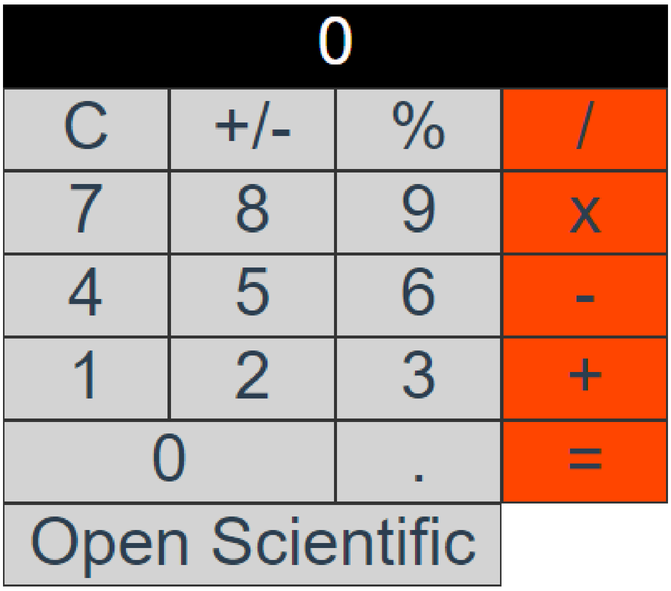

Figure 2. ExampleCalc base calculator view.

Figure 3. ExampleCalc scientific calculator view.

While developing the application, faults were injected. The implemented failures have different detection difficulties. The reporting for the homework task is similar to the lab task. For a full mark, students must find 85% of the failures and fill in the configuration settings table and failure report, where the failure descriptions must be precise. The identification of explicit failure descriptions adds to the complexity, as students are supposed to manually analyze generated reports to narrow down what exactly causes the failures. For that purpose, the JavaScript part of the application is obfuscated.

3.5. Grading

Students can get a maximum of 9 points in this lab and all of the points come from the homework assignment:

- Up to 3 points for describing the settings
- Up to 6 points for failure report (each unique failure is 1,2 points)

Students will lose points if they fail to fill any of the requested columns of the reports. Sample solutions do not give any points, so reporting them is unnecessary. TAs are provided with an example of how to grade the students' answer accuracy because giving explicit failure descriptions could otherwise be inconsistent between different TAs.

4. Lab Execution

The following section gives an overview of how the lab package was used as a part of the Software Testing course and how TAs conducted the lab sessions for that.

The lab "Scriptless GUI Testing with TESTAR" took place on the 11th and 12th of April 2023 as the 7th lab of the course Software Testing. The 121 students participating in the course were divided into six groups and were in different labs.

The lab sessions were divided between 3 TAs, the first one had three sessions, the second had two sessions, and the third had one session. The author participated in two lab sessions with the first TA and interviewed the other two TAs. All of the labs were held on-site in the University of Tartu Delta building. All of the students were using their own laptops to complete the labs. Most of the students used Windows-based laptops, which were provided by the university in collaboration with IT Academy. Roughly 20% of students used Apple products and few students used Linux-based operating systems

The author took part in both of the first TAs sessions on the 11th of April. The lab session was given in the following way: Students were introduced to both ways of how TESTAR could be used: locally on their own Windows machine or using a virtual machine. Then while students downloaded the needed software, the TA followed the slides to show how to set up the TESTAR tool and explained how filters and oracles work in it, at the same time trying to interact with students and teach them how to find the best keyword to use as filters and oracles. As an observation, students who started to download the VirtualBox and the provided .OVA file got the system running in 45 minutes. Also, we ran into an issue where the set filters did not work, but with quick debugging, it was found that while setting filters, no empty spaces should be used, as those broke the regex logic for TESTAR. One student who used MacBook Air with an M2 processor could not run the .OVA file using VirtualBox. Using an alternative virtualization tool, the conversion to the required file format was not successful as the student's laptop ran out of storage space. Another student did not use the provided virtual drive and used TESTAR on his local machine and got TESTAR running. All of the students who stayed till the end of the lab could get the TESTAR running. Also, students taking part in the first TA labs were provided with a USB drive to make the OVA downloading process faster.

With the other two TAs, the author had interviews on the 12th of April. The second TA handled the lab differently and displayed the whole process of TESTAR setup on her machine, ran the tests and caught some failures, and then showed how to fill in the required table for the lab assignment. She used students' input to specify the filter and oracles and used their input to specify how to describe and reproduce the failures for the failure report.

The third TA also followed the slides to show how to set up the TESTAR tool and explained how filters and oracles work in it, but most of the remaining time was used to get students to run the lab assignment tests. The most common issue for students from the last lab group was that spaces or special letters were used in file paths of the TESTAR tool, but those paths needed to be between quotation marks.

5. Feedback

This section gives an overview of the feedback from students. Students were asked to provide feedback in the following lab session, which took place a week after the "Scriptless GUI Testing with TESTAR" lab.

Out of 110 active students, 44 gave answers to the feedback questionnaire. This means that 40% of students provided feedback.

5.1. Feedback Questions

Feedback was asked using Google form, and it consisted of the following statements:

- The goals of the lab were clearly defined and communicated
- The tasks of the lab were clearly defined and communicated
- The instructions of the lab were appropriate and helpful
- The tools used in the lab were appropriate and useful
- Compared to the previous labs, the homework assignment was more difficult
- Overall, what I learned in the lab is relevant for working in the software industry
- Overall, the lab was interesting and inspiring

And in the end, there was a text field with the title "Here you can add additional feedback:".

5.2. Quantitative Feedback

The results of quantitative feedback can be seen in Figure 4. As per the feedback, the vast majority of students agreed that the lab goals were clearly defined and communicated. For 63,7% of the students, the lab tasks were clearly defined and communicated. Only 11,3% of the students disagreed with that, but for 25%, the answer was so-so.

Regarding the statement about if the lab instructions were appropriate and helpful, the overall percentages were very similar to the previous question. 18,2% of students disagreed with the statement: "The tools used in the lab were appropriate and useful" which is a relatively small number as TESTAR is still considered to be a prototype tool and is still under development. But the number of students who did not agree nor disagree is again quite high at 29,5%.

The last statement where the majority of students agreed was regarding the complexity of the homework assignment compared to previous labs. 54,5% agreed with that statement, and 22,7% thought it was more or less similar to previous labs in difficulty.

The last two statements got conflicting answers from students. 27,3% of the students found that things learned in the lab were not relevant for working in the software industry, while 31,8% of students opposed that. Similarly, 40,9% of students thought that the lab was mundane and was rivaled by 43,2% of students who found the lab to be interesting and inspiring.

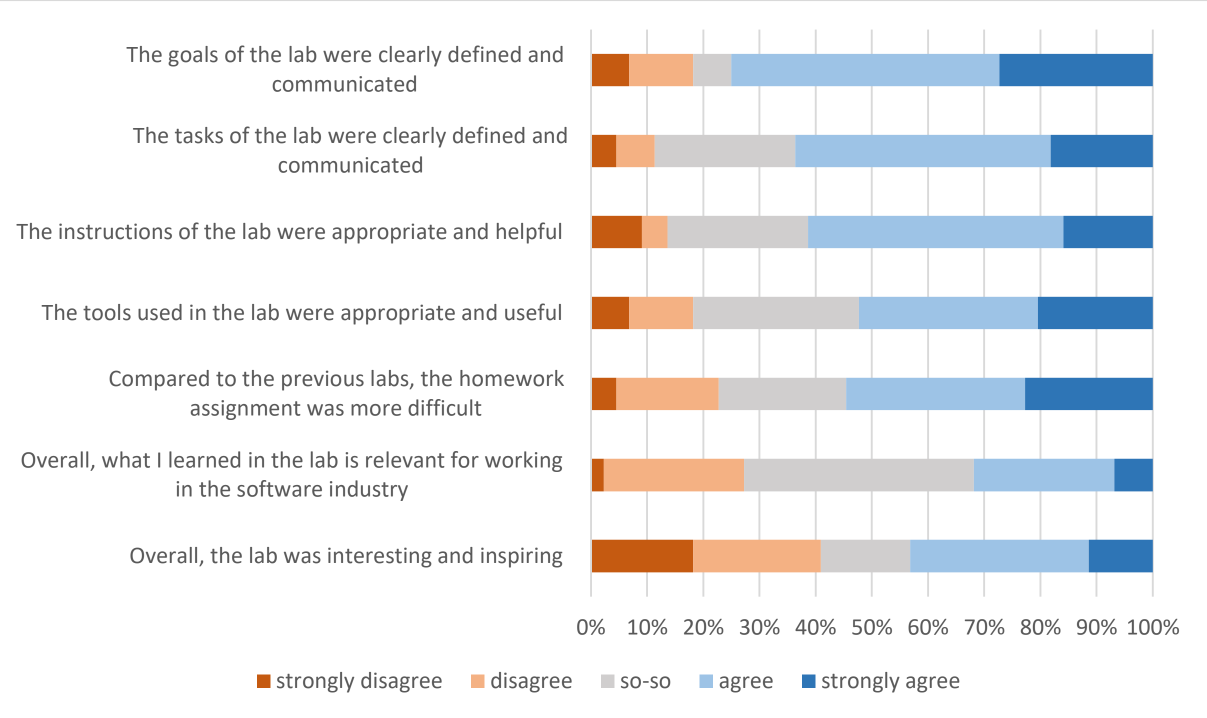

Figure 4. Quantitive questionnaire feedback

5.3. Qualitative Feedback

The feedback tended to be quite contradictory regarding the instructions, where some found the materials to be comprehensive, and a few others found that homework would have needed some additional explanations. On the negative side, students said that there were too many preconditions for the lab, the homework assignment was much more complicated than the lab assignment, and the virtual drive was too big. Also, the lack of examples and outdated official TESTAR documentation were pointed out. On the positive side, some students found TESTAR to be easy to use and the lab package to be interesting.

Students also had some suggestions, like using a USB drive to share the software. As noted previously, this was done in 3 lab sessions, but students still did not prefer that option, as only 3 students out of 15 present in one of those lab sessions used that option. Moreover, a 6th answer option, "No experience in the industry" was suggested for the feedback statement, "Overall, what I learned in the lab is relevant for working in the software industry".

5.4. Comparison to Feedback from Previous TESTAR Lab Package

This sub-section compares the feedback gathered from the previous TESTAR lab package "Automated GUI Testing" [7] and feedback gathered from the current thesis.

As stated by the previous TESTAR lab package author in his thesis, the complexity of the homework should have been higher. And as seen in Figure 5, the difficulty of the homework assignment has increased significantly. Where in 2019, 23,8% of respondents agreed with that statement, but in 2023 this number had risen to 54,5%.

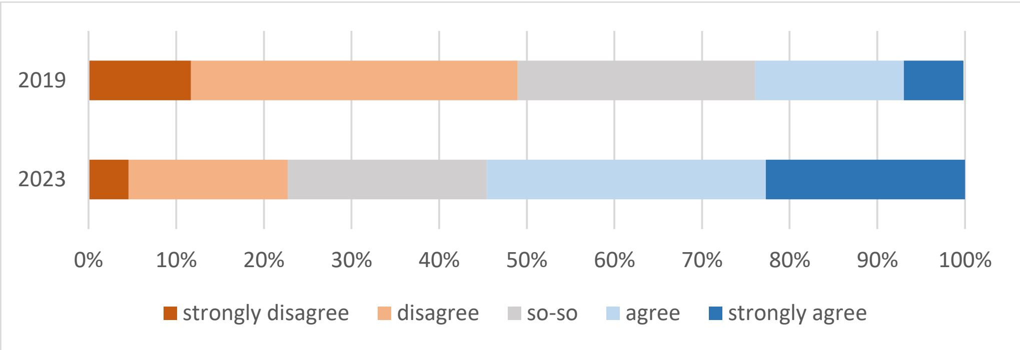

Figure 5. Responses to the statement "Compared to the previous labs, the homework assignment was more difficult" in 2019 [7] and in 2023.

It is quite notable that the number of students who could not decide while answering to the feedback has risen and that far fewer students strongly agree with the majority of the statements.

Table 2. Comparison between students' feedback from 2019 [7] and 2023.

		strongly disagree	disagree	so-so	agree	strongly agree
The goals of the lab were clearly defined and communicated	2019	0,0%	6,8%	6,8%	37,3%	49,2%
	2023	6,8%	11,4%	6,8%	47,7%	27,3%
The tasks of the lab were clearly defined and communicated	2019	3,4%	5,1%	10,2%	28,8%	52,5%
	2023	4,5%	6,8%	25,0%	45,5%	18,2%
The instructions of the lab were appropriate and helpful	2019	3,6%	3,6%	8,9%	41,1%	42,9%
	2023	9,1%	4,5%	25,0%	45,5%	15,9%
The tools used in the lab were appropriate and useful	2019	6,8%	10,2%	17,0%	33,9%	32,2%
	2023	6,8%	11,4%	29,5%	31,8%	20,5%
Compared to the previous labs, the homework assignment was more difficult	2019	11,9%	37,3%	27,1%	17,0%	6,8%
	2023	4,5%	18,2%	22,7%	31,8%	22,7%

Overall, what I learned in the lab is relevant for working in the software industry	2019	3,4%	11,9%	32,2%	42,4%	10,2%
	2023	2,3%	25,0%	40,9%	25,0%	6,8%
Overall, the lab was interesting and inspiring	2019	8,5%	8,5%	30,5%	35,6%	17,0%
	2023	18,2%	22,7%	15,9%	31,8%	11,4%

Table 2 also displays that overall student satisfaction regarding the lab package had fallen as other statements have received less positive responses from the students, where the percentage predominantly differentiated less than 20% from those given four years ago. Most notably, for "The instructions of the lab were appropriate and helpful" the percentage of students agreeing with that statement had fallen more than 20%. Also, in 2019 only 17% of students disagreed with the statement, "Overall, the lab was interesting and inspiring" while in 2023, it had risen to 40,9%. This might come down to the raised difficulty of the homework, as other aspects of the lab package were similar to the previous lab package.

5.5. Future Improvements

As gathered from the student's feedback and interviews with TAs, the biggest issue with the current lab package is the size of the provided virtual machine that is needed. To tackle this issue a smaller version of Windows could be used for the virtual machine, USB drives could be used for distribution in the labs, or a Docker image could be created and used. TESTAR is currently not compatible with two latest Chromium versions 111 and 112, which may not be ideal for students who prefer not to use a virtual machine. However, once TESTAR addresses this concern in its upcoming release, an update will be necessary for this laboratory package.

Additionally, the lab instructions should include a brief introduction on how to utilize commands through the command line interface because a considerable amount of students did not know when and why quotation marks were supposed to be used in the file paths needed in the TESTAR tool setup.

6. Summary and Conclusions

The purpose of this thesis was to update and create student's and teaching assistant’s materials for the lab “Scriptless GUI Testing with TESTAR” used in the Software Testing (LTAT.05.006) course that is taught at the University of Tartu. The lab package was used in the 2022/2023 spring semester by 110 active students out of 121 registered for the course.

All three TAs used their own interpretation of how to conduct the lab, but no significant issues were encountered with the lab materials. Forty-four students provided feedback, upon which future improvement ideas were proposed. The created materials can be used as-is in the future for the Software Testing course or could be improved according to the provided improvement propositions.

References

[1] University of Tartu, Institute of Computer Science. Computer Science curriculum Software Development speciality module. 2023. https://ut.ee/et/oppekavad/informaatika. (31.03.2023)

[2] University of Tartu, Institute of Computer Science. SIS Software Testing (6 EAP) LTAT.05.006 Course Description. 2023. https://ois2.ut.ee/#/courses/LTAT.05.006/version/c82f647c-9c23-0c13-049d-4209e76b4b12/details. (01.04.2023)

[3] University of Tartu, Institute of Computer Science. Software Testing (LTAT.05.006) labs. 2023. https://courses.cs.ut.ee/2023/SWT2023/spring/Main/LabsPracticeSessions. (01.04.2023)

[4] Börjesson E., Feldt R. Automated System Testing using Visual GUI Testing Tools: A Comparative Study in Industry. IEEE Fifth International Conference on Software Testing, Verification and Validation. 2012. https://ieeexplore-ieee-org.ezproxy.utlib.ut.ee/document/6200127 (01.04.2023)

[5] Börjesson E. Industrial Applicability of Visual GUI testing for System and Acceptance Test Automation. IEEE Fifth International Conference on Software Testing, Verification and Validation. 2012. https://ieeexplore-ieee-org.ezproxy.utlib.ut.ee/document/6200141 (01.04.2023)

[6] Vos T. E. J., Aho P., Ricos F. P., Rodriguez-Valdes O., Mulders A. TESTAR – scriptless testing through graphical interface. Software Testing, Verification & Reliability, 2021, volume 31, Issue 3. https://testar.org/wp-content/uploads/2021/09/stvr.1771.pdf. (02.04.2023)

[7] Prink Kert. Lab Package: Automated GUI Testing. University of Tartu, Institute of Computer Science bachelor's thesis. 2019. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=66307&year=2019 . (02.04.2023)

[8] Bons A., Marín B., Aho P., Vos T. E. J. Scripted and scriptless GUI testing for web applications: An industrial case. Information and Software Technology, 2023, volume 158. https://www.sciencedirect.com/science/article/pii/S0950584923000265. (02.04.2023)

Appendix

I. Lab Materials

Lab Materials for Students

- Lab Instructions for Students, PDF file – https://courses.cs.ut.ee/2023/SWT2023/spring/Main/LabsPracticeSessions?action=download&upname=SWT2023L7-Instr-v1.0.6.pdf
- Slides, PPTX file – https://tartuulikool-my.sharepoint.com/:p:/g/personal/taavikar_ut_ee/EU1C642-TLlPibgQXu9O3CEBNq4_YfwkLbdz9CIZBbW6Hg?e=jlyFcA
- Virtual machine (including TESTAR and SUT), OVA file – https://tartuulikool-my.sharepoint.com/:u:/g/personal/taavikar_ut_ee/Ec9QwfcWuyxOg6-1vgeDBgABO9YYkzoBWNzZWu1LJ_fifw
- TESTAR with SUT, ZIP file – https://tartuulikool-my.sharepoint.com/:u:/g/personal/shah_ut_ee/EQ7sFos-j91CmfOU8-1O7EsBYjRyBVWtEnrD_Er5SSimFA?e=jHBYUc

Lab Materials for TAs

- Lab Instructions for TAs, DOCX file – https://tartuulikool-my.sharepoint.com/:w:/g/personal/taavikar_ut_ee/EVA3kO1vzaNLoNZdYLtrFMABm5PeWc7CbXSmJc7viSbHtQ?e=tVcXXI

For confidentiality reasons, Lab Instructions for TAs file is password protected but will be made available on request.

II. License

Non-exclusive license to reproduce thesis and make thesis public

I, Taavi Karelson,

1. herewith grant the University of Tartu a free permit (non-exclusive license) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Lab Package: Scriptless GUI Testing with TESTAR,

supervised by Dietmar Pfahl and Faiz Ali Shah.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons license CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive license does not infringe on other persons' intellectual property rights or rights arising from the personal data protection legislation.

Taavi Karelson

09/05/2023