University of Waterloo

Faculty of Mathematics

Automated Testing for Efficiency and Accuracy

Agriculture and Agri-Food Canada

Harrow, Ontario

Prepared by

Kai Yi (David) Zhang

1B Computer Science

ID 20762627

July 16, 2019

Memorandum

To: PD 11 Evaluators

From: Kai Yi (David) Zhang

Date: July 16, 2019

Re: Work Report: Automated Testing for Accuracy and Efficiency

---

I have prepared the report titled "Automated Testing for Accuracy and Efficiency" for my 1B work term and for Agriculture and Agri-Food Canada (AAFC). This is the first report that I completed for my Bachelor of Computer Science degree in the Co-operative Education Program at the University of Waterloo.

My team at AAFC lead by Jingyi Yang is working on applications that can assist users with creating graphs with data from numerous models. My responsibilities as a Computer Programmer include debugging and adding new features to the applications. This report analyzes the current testing guidelines implemented within the team and how integrating automated testing can increase productivity.

This report is written entirely by me and has not received any academic credit at the University of Waterloo or any other institution. I have not received any assistance other than edits and suggestions from peers and supervisors in the PD11 course at the University of Waterloo. In addition, I would like to thank my peers and supervisors for editing and giving me suggestions for the report.

The Faculty of Mathematics requests that you evaluate this report for quality of technical content, formatting and quality of writing. Following your assessment, this report along with your feedback will be submitted to the Math Undergrad Office for further evaluation by qualified work report markers. The marks combined will determine whether this report receive credit.

Thank you for your support in writing this report,

Kai Yi (David) Zhang

Table of Contents

List of Figures....2

Summary....3

1.0 Introduction....3

2.0 The Testing Guidelines....5

2.1 Issues with the Current Testing Guidelines....6

3.0 EasyGrapher Tester....7

4.0 New Testing Guidelines....8

4.1 Testing Guidelines Benefit Comparison....9

4.2 Testing Guidelines Time Comparison....10

5.0 Conclusion....11

6.0 Recommendation....12

References....13

List of Figures

Figure 1.0. The two main testing components for the basic tests.....2

Figure 2.0. A sample layout for the current testing guidelines.....3

Figure 3.0. A layout for the new testing guidelines.....5

Summary

This report titled “Automated Testing for Accuracy and Efficiency” focuses on the process of integrating automated testing into the current testing guidelines for EasyGrapher DSSAT and automated testing’s effects on the testing results and process efficiency.

The report analyzes the current testing guidelines to identify any potential problems that can influence the testing results. One major problem with the current testing guidelines is related to the manual testing of EasyGrapher DSSAT. Human errors affect the results and create inconsistent data. By analyzing the functionalities of the EasyGrapher Tester, a program used by the development team for debugging, automated testing can be integrated into the current testing guidelines. Using automated testing, the amount of human errors will significantly decrease, which generates more consistent and accurate results. However, automated testing cannot test everything. Although EasyGrapher Tester can assist with the basic tests, developers still need to manually test other features such as interface and user inputs.

The report concludes that the current testing guidelines can provide valuable testing results. However, due to the inconsistency caused by human errors and the time needed for manual testing, automated testing needs to be a part of the testing guidelines. Although automated testing requires the developer to learn the implementation, the efficiency and accuracy of the testing results provided by EasyGrapher Tester are extremely beneficial for the development team.

The recommendation is to integrate automated testing into the current testing guidelines. However, if the developer does not have experience with the implementation of EasyGrapher Tester or programming, using the current testing guidelines is satisfactory. EasyGrapher Tester is a tool to help the development team with testing and should not replace manual testing entirely.

1.0 Introduction

The programmers working with Agriculture and Agri-Food Canada (AAFC) designed EasyGrapher DSSAT. The main purpose of the program is to graph the Decision Support System for Agrotechnology Transfer (DSSAT) output files using Microsoft Excel (Yang, Drury, Yang, Li, & Hoogenboom, 2014). EasyGrapher DSSAT was first developed back in 2002, written in Visual Basic 6 (Yang & Huffman, 2004). Currently, EasyGrapher DSSAT is on version 4.7.5 written with Visual Basic .NET and was released simultaneously with the DSSAT 4.7.5 software.

The DSSAT model is a popular software application. Its purpose is to simulate dynamic crop growth models for over 40 crops. In addition to model simulation, the DSSAT software also offers other utility software which assists users with formatting data on weather and soil (Jones, et al., 2003). EasyGrapher DSSAT is one of the software included with the DSSAT software.

In order to provide DSSAT users with the best experience, the development team must follow testing guidelines to test updates for EasyGrapher DSSAT. The current testing guidelines require the development team to run EasyGrapher DSSAT manually and test different output files from DSSAT. This process can take a lot of unnecessary time which hinders the development team's progress.

The focus of this report is to explore the issues within the current testing guidelines and how automated testing can help eliminate some of these issues. In addition, this report compares the time efficiency and benefits of different testing guidelines.

2.0 The Current Testing Guidelines

The number of graphing options for EasyGrapher DSSAT is immense. There are over 40 crops supported by DSSAT. Each crop has different experiments; each experiment can potentially have up to 19 output files; and within each output file, different data sets can be used as the X-Axis to produce different graphs.

To test EasyGrapher DSSAT, the developer needs to perform the basic tests (Figure 1.0). The developer needs to run EasyGrapher DSSAT on the output files of five selected experiments for five different crops with the default X-Axis (days after simulation). Then, the developer needs to run all datasets as the X-Axis on each experiment's "PlantGro" file (Jones, et al., 2003). The run time of EasyGrapher DSSAT is recorded manually and the resulting Excel file is also saved.

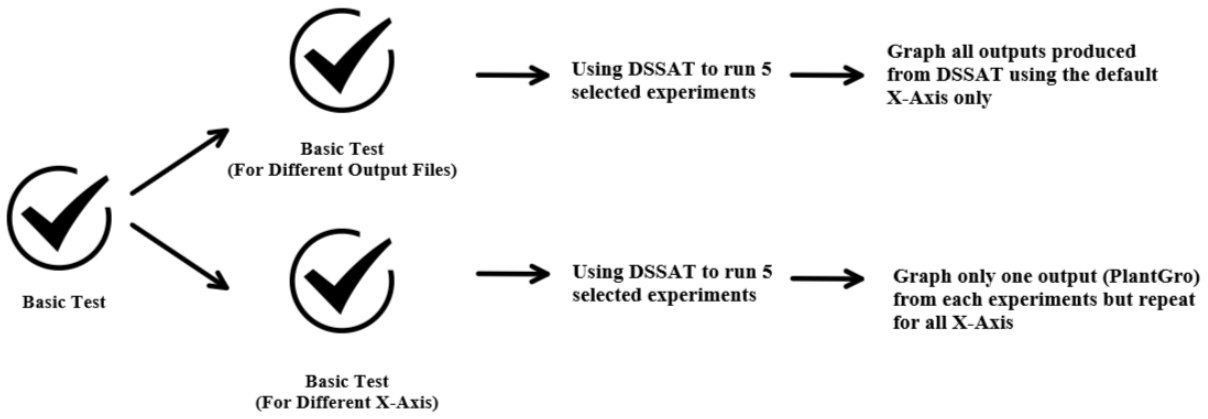

Figure 1.0 The two main testing components for the basic tests. Adopted from *Check Mark Icon #209476* by Free Icons Library, n.d., Retrieved from https://icon-library.net/icon/check-mark-icon-2.html

After the basic tests, the developer needs to run EasyGrapher DSSAT with Microsoft Office in different languages to check for any layout errors. Finally, the newly generated results are compared with older data to ensure the quality of the software update is satisfactory. After the developer has completed all of the steps above, he/she needs to move onto another computer and repeat the entire process again. Based on personal experience, the estimated time for one developer to complete one entire cycle of the testing process is around 4 days (Figure 2.0).

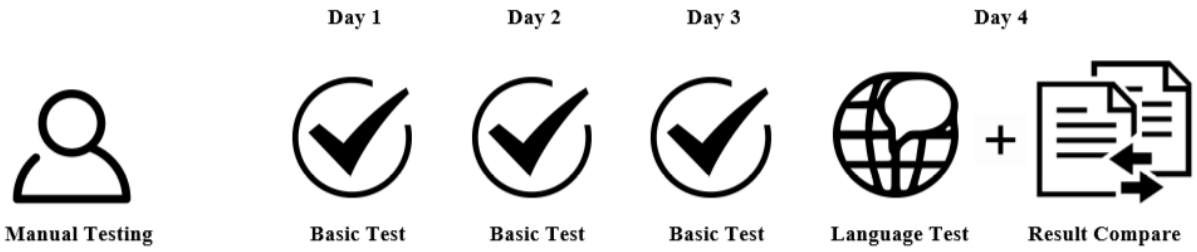

Figure 2.0. A sample layout for the current testing guidelines. Adopted from *Earth International Internet World Wild Planet Language Free Icon* by calumh, n.d., Retrieved from https://www.onlinewebfonts.com/icon/574461; *Account, friend, human, man, member, person, profile, user, users icon* by Enes Dal, n.d., Retrieved from https://www.iconfinder.com/icons/392531/account_friend_human_man_member_person_profile_user_users_icon; *Check Mark Icon #209476* by Free Icons Library, n.d., Retrieved from https://icon-library.net/icon/check-mark-icon-2.html; *Compare* by Mariana, n.d., Retrieved from https://thenounproject.com/term/compare/62588

2.1 Issues with the Current Testing Guidelines

The current testing guidelines can provide useful results. However, there are flaws. One of the biggest problems with the current testing guidelines is the process used for tracking runtime. Using stopwatches to record runtime will yield large margins of error. In addition, there is a lot of room for human error, such as recording the wrong numbers, fatigue from repetitive work and not paying attention to the inputs (Huang, 2017). Furthermore, the reaction time of different developers also affects the results. Two different developers under the exact same conditions will produce different results. In addition, background programs can also affect the runtime, especially if the computing power of the testing computer is not strong. The same test case can produce different results

depending on what background programs are running at the time. A temporary solution to limit the impact of the background programs is to pay close attention to the Task Manager. Overall, it is not recommended for the developers to do too much manual testing as it can lead to inaccurate results and waste unnecessary time and resources for the development team (Kumar & Mishra, 2016).

3.0 EasyGrapher Tester

EasyGrapher Tester is a tool the development team uses to improve EasyGrapher DSSAT. The purpose of EasyGrapher Tester is to automate the testing process and assist with debugging. The development team updates EasyGrapher Tester with new EasyGrapher DSSAT code, which tests files with bugs and identifies errors. Other than helping with debugging, EasyGrapher Tester can also track runtime, save output files, and log errors if EasyGrapher DSSAT crashed. Although EasyGrapher Tester still gets affected by background programs, it is not affected by human errors (Rafi & Moses, 2011). As a result, EasyGrapher Tester is more consistent than manual testing. In addition, by using automated testing, developers can spend more time on other tasks such as writing reports and updating developer and user manuals. On the other hand, since EasyGrapher Tester is automated, it cannot test user interactions and the interface. This means that these tasks must be manually tested. Furthermore, EasyGrapher Tester does not have the option to change X-Axes. The program can only run tests using the default X-Axis. Additionally, the developer needs to spend time to learn the EasyGrapher Tester implementation, which can prolong the testing process.

4.0 New Testing Guidelines

To improve the current testing guidelines, a few changes are expected. First, testing different output files with the default X-Axis must be automated by the EasyGrapher Tester. Second, the developer needs to manually test the interface by selecting different output files from DSSAT to examine whether the graphing options page is displaying the correct options. Third, the amount of testing with different X-Axes must decrease. The remaining steps stay the same as the current testing guidelines. The layout of the new testing guidelines is shown in Figure 2.0. To get the most accurate results, automated and manual testing cannot be completed at the same time on the same computer.

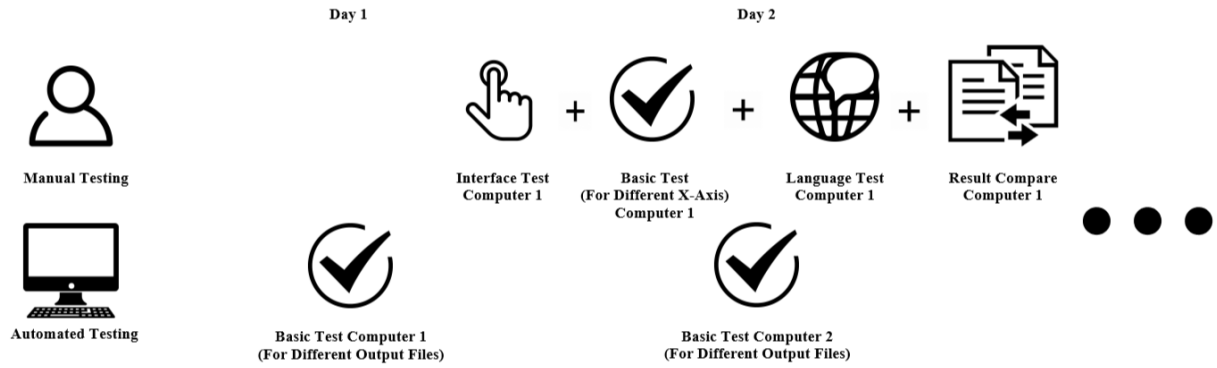

Figure 3.0. A sample layout for the new testing guidelines. Adopted from *Earth International Internet World Wild Planet Language Free Icon* by calumh, n.d., Retrieved from https://www.onlinewebfonts.com/icon/574461; *Account, friend, human, man, member, person, profile, user, users icon* by Enes Dal, n.d., Retrieved from https://www.iconfinder.com/icons/392531/account_friend_human_man_member_person_profile_user_users_icon; *Natural User Interface 2 Icon* by icon8, n.d., Retrieved from https://www.visualpharm.com/free-icons/natural%20user%20interface%202-595b40b85ba036ed117ddbf9; *Check Mark Icon #209476* by Free Icons Library, n.d., Retrieved from https://icon-library.net/icon/check-mark-icon-2.html; *Compare* by Mariana, n.d., Retrieved from https://thenounproject.com/term/compare/62588; *Computer* by Patrick Morrison, n.d., Retrieved from https://thenounproject.com/term/computer/12565/#

4.1 Testing Guidelines Benefit Comparison

To start off, the developer needs to learn how to update EasyGrapher Tester by reading the Developer Manual from the supervisor. Learning EasyGrapher Tester will take a few days, but the knowledge is beneficial. As stated before, automated testing eliminates human errors and provides more consistent results (Rafi & Moses, 2011). In addition, automated testing is faster, as the developer does not need to manually operate EasyGrapher DSSAT (Rafi & Moses, 2011). During the automated testing process, the development team can run tests to check the accuracy of the interface. In the current testing guidelines, interface testing is only a part of the manual process. With the new guidelines, interface testing is the focus of the development team. Compared to the previous guidelines, testing the interface takes much less time, as generating an interface from a DSSAT output file takes less than a second while running the entire program can take half of a minute. By dividing the testing process into two parts, the development team will have time to work on other tasks or test more outputs (Jamil, Arif, Abubakar, & Ahmad, 2016). Finally, by reducing the amount of testing for different X-Axes, developers can deliver more results while preserving the accuracy of the testing process. This is because the datasets that are used as the X-Axes are all produced by DSSAT with a consistent format. The format has not changed and using previous testing results, EasyGrapher DSSAT does not have problems graphing with different X-Axes. However, if DSSAT updates the formatting, the development team needs to pay more attention to the datasets for the different X-Axes.

4.2 Testing Guidelines Time Comparison

The time comparisons between the current and new guidelines are drastic. Based personal experience testing using the current guidelines and using EasyGrapher Tester, if a developer needs to test a new EasyGrapher DSSAT version on four computers, using the current guidelines will require around 16 workdays. If the developer has experience with Visual Basic and the EasyGrapher DSSAT source code, it should only take him/her one to two days to learn and update EasyGrapher Tester. With automated testing, it will only take seven to eight days to test all four computers. Using the new guidelines, the entire testing cycle takes less than half of the time needed for the current guideline's testing cycle. In addition, using automated testing will also reduce worker fatigue. By counting the number of datasets in the DSSAT output files the current guidelines require the developer to run EasyGrapher DSSAT roughly two-hundred-eighty times per cycle. Doing repetitive tasks cause lethargy which will greatly increase the amount of human errors, thus causing more inconsistency in the testing results (Sadeghniiat-Haghighi & Yazdi, 2015). In the case of testing only one computer, where the estimated testing time for the current and the new testing guidelines are the same, automated testing is still preferred for the same reasons as stated above.

5.0 Conclusion

Based on the analysis, automated testing needs to be integrated into the testing guidelines. The current guidelines can provide information about the success and failure of the program, a rough estimate of the running time and a good examination of the interface. However, these testing guidelines require a lot of time, as the number of files and different X-Axes is large. Testing using the current guidelines is satisfactory but spending weeks to get rough estimates and inconsistent results is not an effective way to spend resources.

EasyGrapher Tester, the debugging tool used by the development team, can be used to automatically test EasyGrapher DSSAT. Although it takes a few days to learn EasyGrapher Tester, it is greatly beneficial to development team. Automated testing will eliminate any human errors in the final results and will increase the efficiency in the testing process.

By using the new guidelines and integrating automated testing into the process, the current guidelines will be divided into two parts. EasyGrapher Tester tests part of the basic tests and record the results. The developer checks anything that EasyGrapher Tester cannot test, such as graphing with different X-Axes, interface, languages and comparing the current results with older results. By integrating automated testing, the new guidelines provide an efficient and accurate process for current and future developers of EasyGrapher DSSAT.

6.0 Recommendation

The current testing guidelines needs to include automated testing to provide accurate results efficiently. EasyGrapher Tester is used to test the part of the basic tests with the default X-Axis while the development team tests the interface and graphs with different X-Axes. Other than the changes in the basic tests and interface tests, the rest of the guidelines will stay the same. The development team should continue testing EasyGrapher DSSAT with different languages in Microsoft Office and comparing with previous testing results.

However, if a developer has trouble learning the implementations of EasyGrapher Tester or has no experience with reading source codes, the developer does not need to force himself/herself to learn the implementation of the software. The developer can use the current testing guidelines. The current guidelines do not require the developer to interact with EasyGrapher Tester's source code. However, the developer must note down the time recorded is taken with manual testing as it is affected by human errors.

One final recommendation focuses on the fact that automated testing is only a tool for the development team. Automated testing should not replace manual testing completely as the intended users for EasyGrapher DSSAT are humans and not machines. Although EasyGrapher Tester can run tests faster and achieve more accurate results than manual testers, it cannot test every feature offered by EasyGrapher DSSAT. By following the new guidelines, the development team can efficiently integrate automated testing into their testing process without sacrificing too much time and effort in return.

References

Huang, F. (2017). Human Error Analysis in Software Engineering. *Theory and Application on Cognitive Factors and Risk Management*, 19-29.

Jamil, M. A., Arif, M., Abubakar, N. S., & Ahmad, A. (2016). Software Testing Techniques: A Literature Review. *International Conference on Information and Communication Technology for The Muslim World*, 177-182.

Jones, J., Hoogenboom, G., Porter, C., Boote, K., Batchelor, W., Hunt, L., . . . Ritchie, J. (2003). DSSAT Cropping System Model. *European Journal of Agronomy*, 235-265.

Kumar, D., & Mishra, K. (2016). The Impacts of Test Automation on Software's Cost, Quailty and Time to Market. *Procedia Computer Science*, 8-15.

Rafi, D. M., & Moses, K. R. (2011). Automated Software Testing A Study of State of Practice.

Sadeghniiat-Haghighi, K., & Yazdi, Z. (2015). Fatigue Mangagementin the Workplace. *Industrial Psychiatry Journal*, 12-17.

Yang, J., & Huffman, E. (2004). EasyGrapher: Software for Graphical and Statistical Validation of DSSAT Outputs. *Computers and Electronics in Agriculture, 45*(1-3), 125-132.

Yang, J., Drury, C., Yang, J., Li, Z., & Hoogenboom, G. (2014). EasyGrapher: Software for Data Visualization and Statistical Evaluation of DSSAT Cropping System Model and the CANB Model. *International Journal of Computer Theory and Engineering*, 2010-2014.

Figure Icon Citations

calumh. (n.d.). *Earth International Internet World Wild Planet Language Free Icon.* Retrieved from Online Web Fonts: https://www.onlinewebfonts.com/icon/574461

Dal, E. (n.d.). *Account, friend, human, man, member, person, profile, user, users icon.* Retrieved from ICONFINDER: https://www.iconfinder.com/icons/392531/account_friend_human_man_member_person_profile_user_users_icon

Free Icons Library. (n.d.). *Check Mark Icon #209476.* Retrieved from Free Icons Library: http://chittagongit.com/icon/check-mark-icon-2.html

icons8. (n.d.). *Natural User Interface 2 Icon.* Retrieved from Visual Pharm: https://www.visualpharm.com/free-icons/natural%20user%20interface%202-595b40b85ba036ed117ddbf9

Mariana. (n.d.). *Compare* . Retrieved from Noun Project: https://thenounproject.com/term/compare/62588/

Morrison, P. (n.d.). *Computer.* Retrieved from Noun Project: https://thenounproject.com/term/computer/12565/#

Acknowledgments

I would like to thank Dr. Jingyi Yang, my supervisor, for helping me getting the resources that I needed to complete this report. I would also like to thank, Angela Bi, my co-worker from the University of Toronto, for working with me under Dr. Yang's guidance.