UNIVERSITY OF WATERLOO
Faculty of Mathematics

Pursuing Further Automation in Data Collection

Advanced Micro Devices (AMD)
Thornhill, ON

Timothy Zheng
1B Computing and Financial Management
ID XXXXXX
June 28, 2022

Street
City, Province
Postal Code

June 28, 2022

TBD
PD11 Evaluators
Advanced Micro Devices (AMD)
1 Commerce Valley Dr E
Thornhill, ON
L3T 7X6

Dear TBD,

As we agreed, I have prepared the enclosed report, "Pursuing Further Automation in Data Collection," for my 1B work report and for the TEAM. This report, the first of four work reports that the Co-operative Education Program requires that I successfully complete as part of my BCFM Co-op degree requirements, has not received academic credit.

ROLE IN COMPANY, BRIEF DESCRIPTION OF DUTIES, PURPOSE OF REPORT must be added in a paragraph

The Faculty of Mathematics requests that you evaluate this report for command of topic and technical content/analysis. Following your assessment, the report, together with your evaluation, will be submitted to the Math Undergrad Office for evaluation on campus by qualified work report markers. The combined marks determine whether the report will receive credit and whether it will be considered for an award.

Thank you for your assistance in preparing this report.

Timothy Zheng

Contents

List of Figures

Executive Summary

1.0 Introduction

The purpose of this report is to explore areas in where procedures for data-collection and test coordination could be optimized, and discuss the tradeoff between time invested and time saved in the process of developing and implementing computer-assisted automation scripts in selected areas.

Advanced Micro Devices (AMD) is an American semiconductor company focused on producing high-performance processor technologies for the most demanding use-cases in the modern era (AMD, nd). AMD's portfolio of leading-edge semiconductor products such as CPUs, GPU, FPGAs, and Adaptive SoCs can be found in a broad range of products ranging from consumer electronics such as personal computers (PCs), laptops, current-generation gaming consoles and electric vehicles to enterprise and government deployments of supercomputers that seek to uncover the next leap in meteorological, artificial intelligence, and theoretical physics research (AMD).

In producing these products, AMD requires an extensive understanding on the performance of their products throughout the development cycle. Performance data is used to provide feedback to various internal teams to ensure performance goals are met and to identify areas of improvement. Moreover, data gathered on products near the end of their development timeline is used to prepare marketing assets which are presented to AMD's business partners, or to consumers directly through trade shows and advertisements. In the latter case, performance data is gathered in popular end-user programs such as video games. This report will focus on the exploration of areas in which automation may be implemented or improved in workflows revolving around performance testing on consumer notebook products near the end of their development timelines. Automation takes simple, rudimentary tasks and automates them by digitizing work through the use of tools to streamline and centralize routine tasks (IBM, n.d.).

2.0 Analysis

2.1 Understanding Computer Automation

With the spread of digitally-aided technologies throughout the modern workplace, it seems like a no-brainer for companies and teams to start incorporating digitally aided automation technologies into their daily workflows. The challenges which the author of the report faces in the workplace heavily center around working with repetitive data-heavy tasks such as data ingestion and processing, and can be accelerated exponentially through computer-aided scripts. To provide some context, this problem is far from being isolated in the data sector. “In fact, research indicates that nearly 70% of business processes that could be automated are still manual” (Fleckenstein et. al., 2018, para. 2). The first step to tackling this problem is to identify areas of repetitive, and easily optimized workflows. Once they are found, algorithms and machine intelligence are implemented, speeding up tasks, and eliminating human-error in monotonous tasks, at which point computers are able to identify potential errors and flag them for human review (Fleckenstein et. al.). Cummings et al. (2016) discovered that bright and diligent workers quickly become overwhelmed by tasks that require constant attention and lack dynamism. In reality, tasks such as data collection do require constant attention in case of errors that can break any ongoing collection task, but is incredibly lackluster to observe for long periods of time. In reality, it would be much easier, and less straining on skilled workers if it was possible for computer intelligence to notify a human only if there is a problem, at which point they can use their skills to address and fix the problem. While the automation program is observing data collection tasks, the human worker can spend their time adapting the program to work better, or improve efficiencies elsewhere, ultimately improving worker satisfaction and overall useful output.

2.1.1 Cost-benefit Analysis

It is also important to understand the limitations in working hours and scheduling that arise in the job. It is difficult, and often unreasonable to for example, set aside two hours of the working day attempting to automate a task that may only be used once or twice and is quicker to be done manually. In order to create a program that could be used universally, a more dedicated approach must be taken, which consumes more time and may not see use until near the end of the work term, at which point the tool may be deprecated when presented to the next batch of co-ops working at the same department. Indeed, primary research in this shows that many scripts attempting to automate monotonous tasks exist on our server, but many appear to be convoluted, and lack documentation to where it is often difficult to figure out what they do, nevertheless the procedure on how to use them. It is, therefore, possibly a more practical endeavour to go after the low-hanging fruit in the process. These tasks can be identified through a quick spot-check where if the task could be automated quicker than the time it takes for the task to be done manually once, then it may be something worth pursuing. Once the script for that simple task is built, it may be polished as it is used more, producing a structure for maturing code the more it is used, and leaving ad hoc tasks where an automated script is only required once to be left in its simple form.

2.1.2 Approach

A methodical approach is taken to systematically identify areas for improvement. After reviewing each section of the normal handling of one of the laptops through each stage of the data collection process, it is apparent that the majority of the "low-hanging fruit" present occurs in our data collection and presentation process. Due to the structure of the regular 9 to 5 workday, it is impractical to optimize the installation of prerequisite software, as it is often too dynamic to be reliably automated without unjustifiably high amounts of overhead and sunk costs. To this end, it is also impractical to automate the ingestion of laptops near the head-end of the testing cycle, as it requires a high degree of discussion, planning and logistics work that is best done by humans. These tasks do not align with the issues presented by Cummings et al. (2016) in their report, as these tasks require lots of problem-solving, communication, creativity and dynamism that rule them out of being possibilities for automation. A summary view of a typical workflow through the data collection process is described in Figure 1 below.

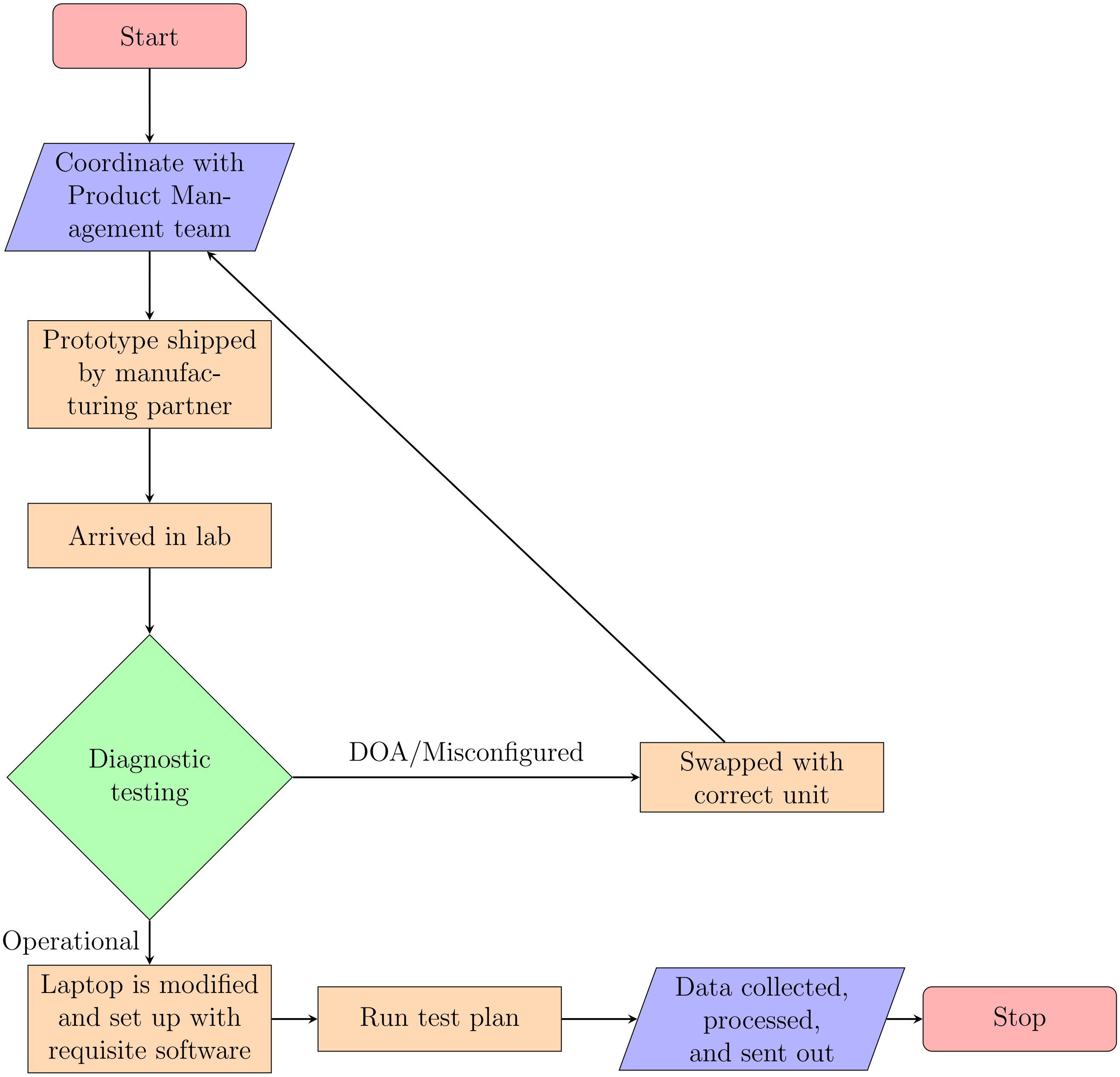
Start
Coordinate with Product Management team
Prototype shipped by manufacturing partner
Arrived in lab
Diagnostic testing
DOA/Misconfigured
Swapped with correct unit
Operational
Laptop is modified and set up with requisite software
Run test plan
Data collected, processed, and sent out
Stop

Figure 1: A typical data collection workflow

Processes in the tail-end of the process outlined in Figure 1 offer many more tasks that could be classified as "repetitive and lacking in dynamism," and are in their nature extremely tedious and time-consuming. This area is therefore ripe for improvements in time and talent utilization. In particular, two areas of interest may produce the most benefit for the amount of dedication required to produce a script to adapt them. If designed correctly, these programs are within the job requirements and if properly documented, could be passed along to future employees with little overhead and learning curve.

2.2 Approaching Potential Solutions

2.2.1 Improving Data Ingestion

Currently, the procedure to ingest data from the team's proprietary game automation program is to manually read and surgically copy-paste successful runs into a central spreadsheet for further data processing. Since certain runs will fail due to a variety of factors that must be individually diagnosed, it is not simple to select an entire range of cells. On average, this process consumes anywhere from 45 minutes to one hour per machine. By matching entries to data points, a simple program could feasibly select just the runs with data and insert them into the spreadsheet in a split-second. Erroneous data points could be then identified and reported to the user for easy re-runs once the underlying problem has been identified and fixed. It can be foreseen that however, such a program will severely dissuade users from proof-reading the processed results and place their faith entirely in the reliability of the program. Unforeseen errors, or data points with data, but outlying results could slip through the script. The development of such a program will require extensive testing, checks and disclaimers when used, but should significantly accelerate what is the largest bottleneck in the process. This task most strongly meets the criteria laid out by Cummings et al. (2016) to be automated due to its tedious and lackluster style, and at face-value appears to have the most promising feasibility to be implemented. The results file produced by our existing testing automation script contains all of the required information to quickly and easily match with entries in our central data processing spreadsheet.

2.2.2 Automatically generating data presentation spreadsheets

When presenting data to data customers for use in product marketing materials, statistical factors need to considered regarding the use of the numbers the lab produces. That is, due to the significant degree of complication of modern processors, run-to-run variance between results is not trivial. It is important to note that at the end of the day, the goal of our testing results is to demonstrate what a typical customer using one of the tested products may expect in a typical workload. It is therefore necessary to run our tests multiple times in order to capture this statistical variance and further process our data points to generate a final result that can be then used by our data customers.

Due to the ever-changing landscape of video games and consumer interests, our coverage of titles generally faces changes on a monthly basis. Each revision requires edits to our data-processing spreadsheets to make space for these changes, which in itself requires significant amounts of time to manually add and remove cells, and edit formulae. Moreover, with the rise in requests for custom test suites from our data customers, it has been increasingly common for these spreadsheets to be made from scratch. Therefore, it would be practical for time to be invested into figuring out some way of automatically producing these spreadsheets to maximize time utilization.

This approach is an excellent back-up option to the former suggested pursuit, primarily because it is significantly simpler, and extremely intuitive to use. Efforts into producing a program to automatically generate these spreadsheets have already started with a simple script that produces the formulae used to process the raw data collected, which already in just one iteration saved time over doing the process manually.

3.0 Conclusions

It may be easy, or rather tempting to suggest changes in all areas, and somehow imagine the development of an "all-encompassing" automation system that could run our entire job at the click of a button. However, the steps required to reach this goal each have their own complexities, and in most scenarios is impossible to practically implement. Ultimately, there will be requirements for each component of the script to be updated as demands shift and new game titles or tests release or are modified. By removing the human factor in these procedures and replacing them with scripted programs, create significant maintenance requirements and abstraction dependencies that are difficult to maintain if a team member is unable to continue maintaining their code. The ultimate pitfall of the idea of implementing automation scripts in every aspect of the job is the fact that it is impossibly difficult to hand off a script to the next user without the burden of writing extensive documentation and modularity that may be difficult to fit in our already tight schedules of testing the machines given to us using our conventional methods.

4.0 Recommendations

The two solutions in the report are not binary choices. If anything, these two solutions are the two most glaring bottlenecks in the lab's typical workday that should work as incentives and proofs of how further automation in the workplace can combat the increase in demand for data and decrease in timeframes currently being faced. Implementing these fixes can lessen the pressure faced by new hires in addition to experienced members of the team by lowering the amount of manual, repetitive, and degrading work (Cummings, et al.). Time that is freed through the use of computer-assisted processes ca be reallocated towards improving the process and reflecting on new potential areas of improvement among other innovative ideas. It is important however, to realize the concerns surrounding the maintenance of these scripts. Those who are responsible for producing these tools much maintain great documentation and instructions on how to use these tools, and, if developed by independent members of the team, should be tasked to those with longer work terms.

References

About AMD — AMD. (n.d.). Retrieved June 12, 2022, from https://www.amd.com/en/corporate/about-amd

Cummings, M. L., Gao, F., & Thornburg, K. M. (2016). Boredom in the Workplace: A New Look at an Old Problem. Human Factors, 58 (2), 279–300. https://pubmed.ncbi.nlm.nih.gov/26490443/

Fleckenstein, M., C. M. O., & Nintex. (2018). The Automated Workplace. Machine Design, 90 (7), 36. https://www.machinedesign.com/automation-iiot/article/21836832/the-automated-workplace

Types of Automation — IBM. (n.d.). Retrieved June 12, 2022, from https://www.ibm.com/topics/automation

Acknowledgments

You may wish to acknowledge people who helped you with your work report. For example, you may thank someone you interviewed or someone whose files you borrowed. Do so in the last paragraph of your letter or memo, or include a separate acknowledgments page where you thank each person and explain briefly how he or she helped.