Reflection Report on PyERT

Team 17, Track a Trace
Zabrain Ali
Linqi Jiang
Jasper Leung
Mike Li
Mengtong Shi
Hongzhao Tan

1 Changes in Response to Feedback

1.1 SRS

- Removed FR3 and Business Event(BE) 1.: After consulting the supervisor, we decided this because there is no sample TUD data provided and the feature is not really important for the product.
- Split FR 1 into FR 1 and 15.: The TA suggested that this functional requirement should be split into two parts, reading and generating.
- Split FR 5 into FR 5 and 11.: The TA said this functional requirement should be split into two parts, generating segments and classifying segments.
- Split FR 8 into FR 8 and 12.: The TA said this functional requirement should be split into two parts, generating and assigning variables.
- Added FR 13 and 14 for extracting transportation network data from the OSM PBF file and OSM API respectively. New system tests have been made for these two FR.: These functional requirements were added after discussing with the supervisor about changes to the program.
- Changed the word 'portability' in NFR MS3 to 'behave the same on different devices' and the description of the NFR: TA said that the original requirement didn't match the correct definition of portability.
- Mentioned about external files that may be used by the packages in the program on NFR SR2: TA said that the requirement should take into account external files from python packages.

- The parts that mentioned R in the Appendix have been removed as R is not necessary to be and has never been used during the developing process of the project.: Supervisor said R was not necessary for the project.

- Deleted traceability matrix as there is no traceability between FR and NFR anymore for this project.: The TA said this matrix was not necessary for our project.

- Changed the fit criteria for FR 5, 6 and 8 and NFR OE1: The TA said that these fit criteria had multiple issues (ambiguous, unverifiable, contradictory, more like rationale than fit criteria)

- Added in fit criteria for FR 15 about what the generated CSV and SHP files should contain.: TA said there was no information about what the required CSV and SHP files should look like.

1.2 Design and Design Documentation

- For the Activity Locations Identification Module, instead of finding the nearest building or amenity for every single GPS point in stop segment(s), find all the buildings that are within a certain distance to the points of the stop segment(s): This change was made after meeting with our supervisor and receiving his feedback for the program we made in Revision 0.

- Moved the functionalities in Main Module for extracting transportation network, buildings, amenities and land-use data from the OSM PBF file or OSM API out to a separate Network Data Utilities Module. New documentation in MG and MIS, Python file for containing the source code and unit tests has been created for the Network Data Utilities Module. This change is based on TA's feedback that the main() function in Main Function Module contains too many functionalities.

- Updated the component diagram in System Design Document to split the merged arrows: The TA said the original component diagram was too confusing and that the merged arrows should be removed to avoid overlaps.

- Updated the Gantt chart and Timeline section in System Design Document by updating the deadlines for each module and including necessary testing: The TA said modules should not be given the same deadlines, should be ordered based on dependencies and priorities, and testing should be mentioned.

- Updated the Services section for Route Choice Analysis Variables Generator Module in MG to make it more clear: The TA said the Services section was too vague.

- Updated the Services section for Main Function Module in MG to make the description more detailed: The TA said the Services section was too vague.

- Added brief explanations of anticipated changes and unlikely changes in MG.: The TA said we should add explanations for each anticipated and unlikely change.

- Updated MIS for Route Choice Generator Module by moving functions that are not externally used into Local Functions and updating the semantics to make it easier for developers to implement according to the MIS: the TA said if the functions not used externally, they should be listed as local functions.

- Updated MIS for Route Choice Analysis Variables Generator Module by replacing the mapLegToStreet function with another function called findNearestStreet: from feedback within the team, we felt that findNearestStreet would be more appropriate for the program.

- In MIS, sorted Abbreviations and Acronyms table to be alphabetical and made capitalization of the abbreviations and acronyms consistent throughout all documents based on TA feedback.

- In MIS, moved filterData and smoothData functions in GPS Preprocess Module to Local Functions section as they are never used externally based on TA feedback.

- In MIS, detailed how detectModes function in Mode Detection Module classifies its modes based on TA feedback.

- In MIS, clarified where each module function receives the inputs from based on TA feedback.

- In MIS, revised formal specification of filterData and smoothData functions based on TA feedback.

1.3 VnV Plan and Report

- Added hyperlinks for the sample data files that were mentioned to be used for the system tests in VnV Plan, based on TA feedback.

- For SRS Verification Plan in VnV Plan, changed 'a group of users' to only 'the supervisor' will be given the system and a set of sample inputs for the system to generate output with the system and given usability survey: We determined this method of testing to be more relevant to our project after from feedback within the team.

- Removed the verification plans in VnV Plan so that the classmates will not review the documents of the system based on TA feedback.

- Specified where the values for PR1 are derived from based on TA feedback and made them more reasonable.
- In VnV Plan, outlined specific inputs for each test instead of just describing it based on TA feedback.

1.4 Hazard Analysis

- Updated Failure Mode, Causes of Failure and Recommended Action for HR1-1 and HR1-2 based on TA feedback.
- Updated Failure Mode and Effects of Failures for HR2-1 based on TA feedback.
- Updated Causes of Failure for HR3-1 and HR4-1 based on TA feedback.
- Added assumptions on user Python version and operating system based on TA feedback.

2 Design Iteration (LO11)

During the design process, emulating the main functions of GERT and generating the same outputs with the same inputs was always the goal. However, the way the program would be implemented has changed over the course of the design process

The original program was supposed to be a modified version of GERT, not a separate program. The plan was to take the original GERT and re-implement all of the functions from the ArcGIS Python package used in GERT with open-source Python packages. That way, the original functionality of GERT could be preserved in this re-implemented version. This plan was created when discussing it with our supervisor Dr. Paez, who had a surface-level understanding of how the program worked.

During our POC demo, we demonstrated how a function from the original GERT program could be emulated using open-source Python packages. At this point, the plan was still to re-implement all the ArcGIS functions. However, when preparing code for this demo, we started to realize that re-implementing the functions would be a tall task. There were multiple reasons why:

- Taking an in-depth look into the code for GERT showed that there were over 100 ArcGIS functions that needed to be remade.
- The documentation for the ArcGIS python package was confusing for the group, since our group wasn't well acquainted with the program and geographic technology.

- The documentation for GERT was not as detailed as we had originally thought. While the main functionality of each module was clearly documented, the code within these modules was confusing and hard to understand. To properly re-implement functions in GERT, the group would have to fully understand the purpose of nearly every line of code.

These difficulties made it hard to create a POC demo, which resulted in negative feedback from the TA and Professor. We realized that going forward, we needed to set a more realistic goal for what the program, to work around these difficulties. We had a meeting with Dr. Paez, and explained the issues we were facing. The conclusion that we came to was that we would create a separate program from GERT that emulated its main functionalities and generated similar outputs, but wasn't constrained to having to recreate GERT line-for-line. That way, we could avoid having to recreate ArcGIS functions, and having to have an in-depth understanding of all lines of code in GERT.

Following this goal, we created our program for Revision 0. This program followed the module structure shown in the original GERT documentation, used similar inputs, and generated similar outputs. The user would input GPS data points, geographic network data, and the number of GPS points to process, and output generated routes, route choice variables, and potential activity locations. The main difference from GERT was that the processing done within each module didn't have to follow the original GERT implementation. In addition to this, we added visualization for generated routes and activity locations using GeoJSON, since we felt that adding this would make the outputs of the program more easily understandable for people who hadn't used GERT before.

Although the feedback from the revision 0 demonstration was mostly positive, we received some feedback for changes we needed to make for revision 1, mostly related to fine-tuning the program. For example, the output of the program for distances between activity locations generated a misleading value. We also demonstrated our product for revision 0 to Dr. Paez, and he also gave some useful feedback. The main piece of feedback he gave was that the proximity of activity locations to the generated routes should be an input from the user.

For revision 1, we implemented changes based on the feedback from the TA, professor, and Dr. Paez. We corrected any misleading outputs pointed out by the TA and the professor. Also, we added an additional input, so the user can choose a radius they want activity locations to be in relative to the generated route. With these revisions, we consider our program to be finished and ready for use.

3 Design Decisions (LO12)

One of the main design decisions we made was modifying our project scope due to limitations with technology. Our initial goal was to modify GERT instead of developing a standalone program that emulates the main functionality of the GERT. However, after research and discussion with our supervisor, we found that this is a tough task due to the reasons that we listed in Section 2 (GERT and ArcGIS are too complex, too many ArcGIS functions). These difficulties made us receive negative feedback for the POC demo, so we set a more realistic goal for our program.

We also decided to use Python as the only coding language for creating our project. One of the main constraints for this product was that it should be open-source and free to use. Python provides many different free-to-use, open-source packages that are well-documented. These packages include technologies that would be relevant to our project, such as GeoPandas and GeoJSON. Another reason we used Python is because we assumed the main users of this program would be previous users of GERT, which also only used Python, so we wanted to keep it the same for the sake of consistency.

Another design decision that we made was to make the program a console program, rather than creating a GUI. Part of the reason for doing this was time constraints, because we wanted to focus the limited time that we had on improving the main functionalities, rather than the aesthetics of the program. Also, we made assumptions that the main need of potential users of GERT would be the main functionality of the program, not the aesthetics.

4 Economic Considerations (LO23)

PyERT is an open-source project implemented with open-source libraries and API. The potential users for PyERT are previous users of GERT, users interested in GERT that couldn't get access to ArcGIS, and our supervisor Dr. Paez. We would attract users to our program by emphasizing the advantages our program provides as an alternative to GERT. The program improves route detection accuracy, provides visualization, is free to use, and is well documented. Since the program is well documented, previous users of GERT who are more familiar with the geography and geographic technologies than the developers could easily contribute improvements themselves.

5 Reflection on Project Management (LO24)

5.1 How Does Your Project Management Compare to Your Development Plan

The project management followed the development plan very well. Each week the team would meet to discuss what needs to be done and tasks to delegate. Each team member would be given a chance to express any concerns or confusion about the tasks at hand. These meetings were largely held on Microsoft Teams. The process of the workflow revolved around Git as planned. Branches would be created and a pull request would be made that other group members must accept in order for the changes to be accepted to master. The group did a good job communicating and creating a smooth workflow to avoid any merge conflicts by carefully delegating tasks and having group calls about any potential changes to documentation or code. The team members had roles that they kept throughout development. This of course did not mean members without certain roles did not have the knowledge to assist those who did.

5.2 What Went Well?

The development plan and overall team communication went well in this project. All team members communicated when they were available to be on group calls and discuss each milestone. If a group member could not make a certain time, team members were very flexible and willing to change meeting times to accommodate any members. Group members were more than open to do or help any member with any task to efficiently advance the project. The open-source free libraries that Dr. Paez suggested worked really well to emulate the functionality of GERT. There is a lot of documentation and examples available online to understand how these libraries work. The software was designed to be very modular which meant it was fairly easy for group members to test the functionality of each module. This meant group members did not have to wait for the implementation of one module to be able to test other modules as long as there was an understanding of what were the sample outputs of each module. Each group member was also experienced in the software development process and as a group was able to complete each document smoothly with little confusion.

5.3 What Went Wrong?

One issue from a technological point of view was the inability to run the original toolkit GERT. Dr. Paez sent the source code of the toolkit but it was not documented well enough to understand the functionality of each script to re-implement it in Python. In numerous attempts to run GERT, the group was not able to get the toolkit fully working and after notifying Dr. Paez the group still could not get GERT running. This led to a lot of confusion early on in the project as well as more time being spent trying to understand the use cases of GERT. Another issue from a communication standpoint was the lack of meetings with Dr. Paez early on in the development lifecycle. Often times the group would find concerns or issues that needed to be clarified by Dr. Paez, however, the group would communicate too late before a milestone and Dr. Paez would not be able to get back with the answers in time through no fault of his own. This led to a lot of assumptions and confusion early on in the development process.

5.4 What Would You Do Differently Next Time?

For the next project, we will do research on the technologies we will be working with well before the proof of concept demo, to determine the different limitations of these technologies. As mentioned in section 2, the most difficult stage of working on our project was when we were trying to follow our original plan of re-implementing all functions of GERT and ArcGIS. Lots of time was wasted trying to figure out how to use these technologies. It was hard to pivot our idea because of the deadlines we faced. If we had originally done more research on these technologies beforehand, we would have been able to pivot easily well before the POC.

Another thing we will do differently for the next project is trying to have a more organized meeting schedule with the supervisor. As mentioned in section 5.3, the group had trouble scheduling meetings with Dr. Paez, who would be able to provide useful feedback about the program since he is one of the main stakeholders. This experience shows that regular communication with important stakeholders needs to be established at the beginning of the design process, to avoid situations like these in the future.