**SEGP2 Group 3 Milestone 4 Report**

**Total Hours Spend:** approx. 154 hours

# Executive Summary

Milestone 4 began by reviewing the findings from Jian Huang's research into various plugins frameworks adopted by other software projects. The decision to take the path of converting the existing Earth plugins architecture (code inclusion) to the proposed API (Application Programming Interface) plugin architecture. The group had agreed not to work any tickets as this task is expected to be heavy and will relinquish all the budgeted hours.

Both Huang and Ken took up the task for laying the groundwork for the API framework. Since this is supposedly a continuation from Milestone 3, and Ken were not present during Huang's presentation on the proposed API framework, both of them had spend most of the time trying level out expectations on the tasks taken. Unfortunately, constant communication failures, and personal and academics commitments had plagued this pair from moving forward as quick as expected. Ken had expressed that he had spend about 53 hours in this milestone, while Huang had reported that he had spend an estimated 82 hours for this sprint.

While both the MSE members were busy working on the API framework base, the BE(SE) members worked on other aspects of the project. Jon had volunteered to take over the administrative tasks of the group. Besides that, it was planned to have both Jon and Sahil to explore the front-end of Earth and try to get the same API concept applied onto the front-end. However, Sahil "isolated" himself from the group and was not able to provide progress updates to the group, which forced Jon to work alone, while consulting both Ken and Huang when doubts arises. Overall progress from Jon had been good, where he already had an idea what and how should the front-end API should be done. Jon had reported that he had spend 19 hours in this sprint, inclusive of all the hours taken for administrative work. Unfortunately, the group does not know how much time Sahil had spend on any aspect of this project, as of the time of this report is written.

Milestone 4 can be considered as another unproductive milestone. This is because there is no solid plugin prototype developed and presented, which was the expected deliverable during planning time. The following are the key takeaways of this sprint for this group:

- **Lack of member commitments:** It was observed that other academic commitments (research project and course assignments) and personal commitments (waged work) had affected the overall performance of the members of this group. For one case, no work has been done at all.
  - Plan: To stress the importance of this project to be delivered back to a real customer (RSP) and lack of commitments will not only discredit ourselves, also the school in general.
- **"Do more with less" is difficult to achieve:** It was planned and discussed that the we, as a group, will not be affected by the decreasing number of head count. The group agreed to strived to do more with less, but not committed after each member had been swamped with other commitments. Also, this strategy is related to the previous takeaway.
  - Plan: Put forth this problem to the full group and had it solved by everyone by the upcoming meeting.

# Individual Reports

Attached are the individual reports of each member of the group.

**Ken S'ng Milestone 4 Report**

# Executive Summary

I have self-assigned myself to work with Jian Huang on the plugin framework, as proposed by himself at the end of Milestone 3. I have spend about the first 10 hours trying to brainstorm with him on how this proposed framework work. Due to communication issues, it required me unexpectedly more time to learn about the idea. Besides that, I have also spend time trying to catalogue the school computers for our use. And due to an unexpected software issue, the troubleshooting took away quite an amount of time away from doing actual work. At the end of this milestone, a redesigned plugin framework surface was delivered.

# Tasks and Activities Assigned

- Create API standards base on the functionality of the file monitor.
  - Sub-task 1: Look at the file monitor system and list (extract possible API methods needed.
    - Description: Prior to this task, it took me 8 hours to try to understand Jian Huang's proposal. The long time taken was due to communication breakdown between both us, as he had troubles articulating his thoughts clearly to me which caused a lot of confusions and misunderstandings. Another 5 hours were taken up to polish up my RoR skills after the semester break, and trying to fully understand how the framework can be "imported" like Java, and be sub-classed/instantiated for use.
    - Idea/Solution: N/A
    - Affected files: `earth_plugin_layers.pdf`
    - Git commits: `(segp2) 1de2dfee0b33a8d66198e10960db50fa8984fd4b`
    - Estimated time taken (planned): 16 hours
    - Estimated time taken (actual): 14 hours
  - Sub-task 2: Develop (code) the API itself (just for file monitor).
    - Description: Initially, we planned that we are going to give the most hours for this task. However, due to personal and other academic commitments, the hours were not met. During this task, a lot of issues were encountered. The language barrier that I had with Huang, coupled with our naivety about how the "require" keyword is used cost us 16 hours to get the initial prototype working.
    - Idea/Solution: N/A.
    - Affected files: `file_monitor.rb`, `api_eta_printer.rb`, `api_file_monitor.rb`
    - Git commits: `(pamalite/earth) 2d087d82d8f117678b75a6a70efdad435fc0f8c4`
    - Estimated time taken (planned): 62 hours
    - Estimated time taken (actual): 16 hours
- Replace what is current available on file monitor.
  - Sub-task 1: Re-develop file monitor with the proposed API.
    - Description: This task was done in the previous task.
    - Idea/Solution: N/A
    - Affected files: N/A
    - Git commits: N/A
    - Estimated time taken (planned): 64 hours
    - Estimated time taken (actual): 0 hours

  - Sub-task 2: Publish standards and API
    * Description: This task was not taken due to an absent of a proper functional prototype to be documented.
    * Idea/Solution: N/A
    * Affected files: N/A
    * Git commits: N/A
    * Estimated time taken (planned): 1 hour
    * Estimated time taken (actual): 0 hours

- Cataloguing SEGP2 computers
  - Description: This task was self-assigned to create a central repository of all the computers provided by the school. It was undertaken due to the fact most of the members have trouble locating the machines. The catalogue was created on the SEGP2 forum, and some FAQ written to give specific answers to very frequently observed issues face by myself.
  - Idea/Solution: N/A
  - Affected files: `http://forums.cs.adelaide.edu.au/mod/forum/discuss.php?d=7243`
  - Git commits: N/A
  - Estimated time taken (actual): 1 hour

- Setting up VNC for remote desktop purposes
  - Description: This task was self-assigned to mitigate the issue of an unexpected locked lab. It took me quite a few of my time to figure out how to tunnel VNC protocol through SSH within the school LAN.
  - Idea/Solution: N/A
  - Affected files: `http://forums.cs.adelaide.edu.au/mod/forum/discuss.php?d=7254`
  - Git commits: N/A
  - Estimated time taken (actual): 8 hours

- Setting up Aptana IDE on SE1.
  - Description: This task was self-assigned as it has something to do with my on-going research project to allow developers to log their time without actually logging them. However, this experiment was met with a disaster after it was revealed that Aptana might had reconfigured the underlying Rails library. Most of the time was spend on troubleshooting the issue.
  - Idea/Solution: N/A
  - Affected files: `http://forums.cs.adelaide.edu.au/mod/forum/discuss.php?d=7243`
  - Git commits: N/A
  - Estimated time taken (actual): 10 hours

## Resource Contributions

For this milestone, I have estimated that I had spend about 53 hours (including 4 hours meeting time) over the spread of 3 weeks. I have only a budgeted 45 hours (15 hours per week) for this semester. The other 15 hours were spend on my research project.

Looking at the hours I felt that I have not had a productive 53 hours. Most of the hours were mainly spending on troubleshooting and leveling language barriers with my task partner.

## Rooms for Improvement

- Better time-estimations: Failure to estimate time required a particular tasks had always been a setback for the group, besides myself. Since this is the first time the nature of the tasks were different from the previous milestones, it takes some time to get myself into my comfort zone. I am aiming to give better time estimates in the coming last few milestones.
  - Plan: Review more on past milestones, and perform reality checks during planning.
- Better communication with other members: It was almost impossible to communicate with my task partner due to language barriers. I had to resort to language translations and lengthy explanations to get a simple thought across. My aim for the coming milestones is to improve my way of "internationalizing" my thoughts as efficiently as possible to get the discussions moving as fast as possible.
  - Plan: Level communication expectations between my task partner and I before any discussion take place.

**Jian Huang Milestone 4 Report**

# Executive Summary

Research on Plugins. Aftering semester 1, I found almost plugin system use APIs, but, so far, most of us have done some tickets, which means that if using APIs, we need to redo what we've done. Therefore, I compare some other skills. Two main techniques of them are Method 1: Use code, which can run itself and Method 2: APIs.

# Tasks and Activities Assigned

- Method 1 Use code.
  - Sub-task 1: Look at how it works in ruby on rails.
    - Description: The function can be used in any tickets. When tickets are installed in earth, the function will run firstly and then trigger the main functions in the ticket. So ticket can run itself.
    - Idea/Solution: Search wheter there is function like other languanges, Java or C++, that can use directly, or we need to build it ourselves.
    - Estimated time taken (planned): N/A hours
    - Estimated time taken (actual): 40 hours
  - Sub-task 2: Solve the security issue.
    - Description: There is a security issue in this way, and some tickets have mentioned that we need to think about which part of earth coding we can expand and / or extend.
    - Idea/Solution: Set privilege, or protect function to avoid illegal action.
    - Git commits: Spend lots of time on it, and, unfortunately, have on idea about it.
    - Estimated time taken (planned): N/A hours
    - Estimated time taken (actual): 20 hours
- Method 2 APIs.
  - Make structure on earth.
    - Description: Set earth structure and express how other ways are not good enough to solve our problems.
    - Idea/Solution: Discuss it with other group member to get same idea and answer their questions.
    - Affected files: N/A
    - Git commits: N/A
    - Estimated time taken (planned): N/A hours
    - Estimated time taken (actual): 5 hours
  - Sub-task 2: Publish standards and API
    - Description: Set three main parts up-level: front-end, back-end, database
    - Idea/Solution: rebuild file_monitor
    - Affected files: N/A
    - Git commits: N/A
    - Estimated time taken (planned): N/A hour
    - Estimated time taken (actual): N/A hours
- Rebuild file_monitor

  - Description: Set an example on how to use APIs to create plugin.
  - Idea/Solution: Divide it into several basic functions
  - Comments: Divide file_monitor into several defferent basic functions, no matter which of privilege it is in the file_monitor. However, I change mind again, when nearly finishing, and use the next skill). Set file_moniter as a up-level package to control database. Do the new code with ken and test it.
  - Estimated time taken (actual): 15 hours

- Discuss with other MSE students
  - Description: See whether we need to rebuild subgroup to do our MSE project.
  - Idea/Solution: Make us more effectional and more ambitional
  - Comments: Most of us eager to rebuild.
  - Estimated time taken (actual): 1 hours

- Research on the Gui.
  - Description: Ruby is different from Javascrip and Asp, and observer part can meet problem to use APIs.
  - Idea/Solution: Talk with friends, search info from webpage.
  - comments: Find problem, try to solve it.
  - Estimated time taken (actual): 1 hours

## Resource Contributions

No idea how to write this part.

## Rooms for Improvement

I need to estimate time and try to show more what I have done, especially coding.

**Jonathan Velasco Milestone 4 Report**

## Executive Summary

I took responsibility for the administrative tasks and elaboration of agendas and minutes. Also, I was originally, together with the rest of the group, assigned to work extracting the possible API methods, however, this task was then taken by Ken and Jian. And I was suggested by Ken to start studying the GUI together with Sahil.

## Tasks and Activities Assigned

- Create API standards base on the functionality of the file monitor.
  - Sub-task 1: Look at the file monitor system and list (extract possible API methods needed).
    - Description: I start by looking at the file monitor to try to extract the possible API methods. I spent approximately 3 hours looking at it. However, I needed to discuss it with Ken in order to get a better understanding of how to extract the methods for the API
    - Idea/Solution: N/A
    - Affected files: N/A
    - Git commits: N/A
    - Estimated time taken (planned): 2 (just looking how to extract few methods in order to check I was going in the right track hours
    - Estimated time taken (actual): 3 hours
- Study GUI thinking on how to implement Jian idea to it. I decided to focus first on the TABS. I splitted it into two tasks
  - Sub-task 1: Creation of a TAB (investigation)
    - Description: The idea that I have is to create a system to be able to create a new tab automatically. I looked at how Ken created the UserUsage tab, and broken down into the essential components. I check the file that he created/modified in order to create the tab. Doing this took me approximately 3 hours.
    - Idea/Solution: N/A
    - Affected files: N/A
    - Git commits: N/A
    - Estimated time taken (planned): 1 hours
    - Estimated time taken (actual): 3 hours
  - Sub-task 2: Automatic create of TAB
    - Description: After finding the files needed to create a tab, I started searching and testing (by trial and error), if it was possible to create methods using rails to read all the files (modules) that are inside a folder. This idea was to create all the files needed for the tab and store them in a folder, and then just 'include' those files (modules). However, no usefull information was found which helped me to do this. This took approximately 4 hours. Note: a latter idea came to me at the end of the milestone and only had time to mention it to Ken but we have not discus it yet. I am thinking to use the idea behind the ruby plugging system to create an automatic way to make tabs. The idea is to create a script (something like scrip/makeNewTab tab_name) which will ask the user for the name of the tab and then, it will automatically generate the files need for the tab (as I think all the

files need to display the name of the tab in the browser have the same headers and starting information) and then just tell to the user on which of the newly create file he has to place the code that perform the function related to the tab. In other words, this idea of automatically create a tab is just create and empty tab with the given title.
    - Idea/Solution: Have not yet try the latest idea
    - Affected files: N/A
    - Git commits: N/A
    - Estimated time taken (planned): 2 hour
    - Estimated time taken (actual): 3 hours

- Update of the ticket in trac
  - Description: This task was assigned to me by Ken. He asked me to modified/update the tickets information on the trac system
  - Idea/Solution: tickets where updated
  - Affected files: tickets 66 and ticket 23 on earth trac system
  - Git commits: N/A
  - Estimated time taken (planned): N/A
  - Estimated time taken (actual): 0.5 hour

- Elaboration of documents
  - Description: Elaboration of agendas and recording of minutes
  - Idea/Solution: N/A
  - Affected files: `agenda_200808071000.pdf ,agenda_200808121100.pdf, minutes_200808071000.pdf, 200808131100.pdf`.
  - Git commits:
    - `(pamalite/segp2) afa445905b3c14c36af3b8d31741091b7ada694b`
    - `(pamalite/segp2) 2d02ba30c05b46833d8cb2d97054fe86ad2c6475`
    - `(pamalite/segp2) 603ca2bc3aeb68fee990eb76f6c18b6edfc706ec`
    - `(pamalite/segp2) 2dc1a2d95431580dcd84329db501f5983964de3f`
  - Estimated time taken (planned): N/A
  - Estimated time taken (actual): 0.5 hours for agendas, 1 hour for minutes

## Resource Contributions

For this milestone, I have estimated that I had spend about 19 hours (from which 7 hours where on meetings, 3.5 hours where administrative tasks) over the spread of 3 weeks. I have only a budgeted 18 hours (maximun 6 hours per week) for this semester.

I think the investigation that I conducted did not produce results more than 'things that will not work', which made me feel disappointed as I was expecting to end this milestone with a first working prototype of how to at least create tabs (empty but with a title) in the browser in an easy way. I think the problem is my knowledge of ruby and rails, on which I basically do a largely and time consuming 'trial and error' instead of know what I am doing.

## Rooms for Improvement

- Do not underestimate the time that I need to understand and work with ruby on rails
- Keep in touch with my partners in order to ensure am in a right track or to check if they know how to solve any problem that I may be stuck with on work