## Process Overview:

1. Decided goals needed in order to produce MVP by the end of this phase.
   1. Each goal/task was segmented into individual components and assigned to individual members.
2. Intermediate deadlines were set for each the group member.
3. Scrum meeting were held every 3—5 days to evaluate progress and determine next steps as well as adjust the schedule in case of any delays.

## Initial Planning and Beginning of Process:

The previous phase provided an excellent backbone structure to build off of for the second phase of the project. Main functionalities had already been implemented and incorporated into the website, such as:

* Visual Tree Diagram
* Compiler
* Basic Structure of Game
* Basic Layout of Website

As such, for this phase, the group decided to focus on the following goals as they were most detrimental towards developing a polished and complete MVP:

1. Developing a fluid user experience
2. Creating content nodes to be added to the learning trees
3. Adjusting any bare bone implementations to something more specific (e.g., the compiler exercises, games, etc.)

## Organizational Strategy

Based off of successes from the previous phase, the group chose to use the Scrum method of meeting and planning in order to efficiently complete the process. The Scrum method of meeting works well for the development of the product because it allows for the creation of well-defined intermediate tasks and is flexible enough to accommodate for any delays/changes that must occur during the development process.

**Definition of Scrum Master:** The scrum master is responsible for taking notes at each of the meetings and recording the progress of the group. The scrum master should also analyze the team meetings to synchronize and adjust plans as necessary.

**Adjustments to Scrum Process**:

* (Kept from previous phase) No product owner, entire team takes responsibility of product owner position
* (Kept from previous phase) Periodic update meetings every 3-5 days run by scrum master
* (Not kept from previous phase) Members no longer have to keep an individual update file.
  + It is redundant considering that the commit feature on github. Additionally, these weren’t referenced often and provided little advantage to the product development.

Additional tools the group used to facilitate the organization of the project:

* **Skype**: provided efficient means of communicating with multiple people at once, organizing meetings, and leaving messages for other team members to be read at a later time.
* **In-person meetings**: Beneficial for demonstrating individual progress, explaining concepts to struggling members, deciding next steps and adjustments
* **GitHub**: Workflow included solely committing to master (major flaw – more info in review and retrospective)
  + Issues were used towards the end of the phase (major flaw – more info in review and retrospective)

## Sprint Backlog

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Major Component | Task | Responsibility | Size | Monday, March 7th, 2016 | Friday, March 11, 2016 | Monday, March 14, 2015 | Wednesday, March 16, 2016 |
| User login system | Reformat Database to Store Exercises and Lessons | Allen | L | 0 | 0 | 8 | 8 |
| Implement the store submissions route | Allen | L | 0 | 0 | 8 | 8 |
| Tree Nodes & Content Pages | Introduction to C Syntax | Shamama | S | 0 | 0 | 3 | 0 |
| Variables Node | Shamama | S | 0 | 0 | 0 | 3 |
| Pointers | Flora | S | 0 | 4 | 2 | 0 |
| Memory Allocation | Flora | S | 0 | 0 | 0 | 3 |
| Exercises | String matching exercises using JavaScript | Shamama | M | 0 | 0 | 0 | 4 |
| 3 Pointers Exercises | Wenfeng | M | 0 | 0 | 3 | 1 |
| 2 Pointers Games | Justin | L | 0 | 2 | 8 | 8 |
| User Interface | Styling the page – Css and JavaScript | Shamama | M | 0 | 4 | 0 | 0 |
| Flora | M | 0 | 4 | 0 | 0 |
| Allen | M | 0 | 0 | 0 | 2 |
|  | Compile code through JQuery Ajax.post | Wenfeng | M | 0 | 0 | 4 | 0 |
| Test Cases | Test Cases for 3 Pointer excercises | Wenfeng | M | 0 | 0 | 4 | 4 |
| Merge | Justin’s game merge to Master | Allen | S | 0 | 0 | 0 | 1 |
|  |  | **Total:** | 73 | 0 | 18 | 40 | 42 |

### Estimation of Task:

The sizes of tasks were decided according to four basic criteria:

1. How many components would be required for that tasks (i.e., many different languages, many functions, etc.) – The more components, the harder the task
2. How dependent the task was on previous, already implemented features. The less dependent it was, the harder the task.
3. The experience level of the member assigned to the task – The less experience the member had with the language, the higher the difficulty.
4. The expected length of time for each task.
   * Small tasks are expected to take around 3 hours of focused, undistracted work
   * Medium tasks are expected to take around 5 hours of focused, undistracted work
   * Hard tasks are expected to take around 6 hours of focused, undistracted work

## Update Meetings

Meeting One: Date: March 7, 2016, 2:00-3:00PM | Skype | Attendance: Everyone

* Came up with tasks for this phase of project.
  + Want to finish most of the implementation so that the last phase only has content to add/stylistic changes to make
  + Gamification features will be left for last phase with a small minority implemented this phase
* Divided tasks amongst members

Meeting Two: Date: March 11 2016, 5:00-6:00pm | BA3200 | Attendance: Everyone

* Everyone presented what they had done and what they had worked on:
  + Flora, Shamama – working on basically the content of their respective nodes.
    - Will both have at least one node done for Mondays meeting completed with HTML, CSS, JavaScript components.
  + Justin – Looking into how to adapt his game
  + Allen and Wenfeng working on multiple backend secions

Meeting Three: Date: March 14 2016, 4:00-6:00pm | BA3200 | Attendance: Everyone

* Allen and wenfeng have implemented compiler exercises to add on to node pages
* Flora has pointers page complete
* Shamama is behind on both her pages
  + Pushed deadline of both to Tuesday
* Justin is trying to make it such that his game resizes in browser

Meeting Four: Date: March 15 2016, 9:00-10:00pm | Skype | Attendance: Allen, Flora, Shamama

* Shamama has submitted both her nodes.
* Flora has submitted both her nodes
* There are multiple style sheets that conflict with each other.
* Need to merge/find solution/undo overwrites

Meeting Four: Date: March 16 2016, 5:00-9:00pm | Skype | Attendance:Everyone

* Finalizing project, going over report
* Checking issues and closing issues
* Resolving errors with styling

## Review and Retrospective

### Successes

1. Using issues and having other members confirm closed issues

Issues were used towards the end of the development process. Despite this, they provided major insights into the project’s progress. Additionally, since members were able to see which features were fully implemented/completed, they could review and test them. This became very advantageous in finding bugs in any completed portion of the code.

* + An example of this: https://github.com/csc301-winter-2016/project-team8/issues/17

1. Scrum meeting – member updates and explanation of code results in efficient progression towards new tasks.

Again, keeping Scrum process with the group’s adaptation provided to be extremely useful for the development of the C Programming Foundations website. Particularly, presenting and explaining completed code saved a lot of time in trying to understand the code. Additionally, the meetings provided help to those that were struggling with their individual components. Both these features significantly improved the efficiency of the development of the product.

### Improvements

1. Issues should be created at the beginning of the initial planning process.

The group was not aware of issues in GitHub. Once they were created, they provided an excellent way of keeping track of what was completed. It also allowed for the definition of strict requirements and prevented issues with multiple people committing the same work/

* + An example would be the style issue: done by multiple people, did not work across all pages

1. Need to use new GitHub work flow that doesn’t override other member’s commits

The GitHub workflow that the group is currently using is not the best method of committing and doing work. The current workflow just includes committing to the master. However, there were occasions in which work was overwritten and had to be recovered. This lost the group a lot of time and caused unnecessary delays. For this reason, the GitHub workflow will be changed in the next phase so that it will have each member fork their own copy and pull request will have to be merged.

* + An example: <https://github.com/csc301-winter-2016/project-team8/commit/c5b1c5ea1d4761ec35e6e71bf7f232a997901f75>