|  |
| --- |
| ClubUML |
| Lessons Learned |
| Navigation/GUI/Systems Team |
|  |
| Ken Canaan & Thomas Cashavelly |
| 3/26/2014 |

|  |
| --- |
|  |

# Team

## Ken Canaan

Ken has the strengths of:

- Project Management

- System Design and Analysis

- Web Design and Progamming (HTML, CSS, JavaScript)

- JAVA (skill level: novice-intermediate)

### Tasks

Currently right now there are 20 tasks within Jazz Hub that are allocated with Ken, with 5 of them still open. The efforts within each tasks range from research, system design, development and documentation. A report of all tasks by Ken can be seen below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Id | Summary | Owned By | Status | Planned For |
| Epic | 15369 | Navigation | Ken Canaan | In Progress | Backlog |
| Task | 22377 | Modify Existing js, jsp, html files for css references | Ken Canaan | New | Sprint 4 |
| Task | 22376 | Consolidate CSS code implementation | Ken Canaan | New | Sprint 4 |
| Story | 22375 | CSS Consolidation | Ken Canaan | New | Sprint 4 |
| Task | 22369 | Navigation Lessons Learned: Ken | Ken Canaan | New | Sprint 4 |
| Task | 19773 | Create Super State Diagram for all ClubUML application | Ken Canaan | Done | Sprint 3 |
| Task | 19767 | Create UML Activity Diagram of Navigation | Ken Canaan | Done | Sprint 3 |
| Story | 19766 | GUI Design - Documentation | Ken Canaan | Done | Sprint 3 |
| Task | 18757 | Port Tab-Menu to manageRelation.JSP and Create Header | Ken Canaan | Done | Sprint 2 |
| Task | 18756 | Port Tab-Menu to refineClass.JSP and Create Header | Ken Canaan | Done | Sprint 2 |
| Task | 18755 | Port Tab-Menu to selectClass.JSP and Create Header | Ken Canaan | Done | Sprint 2 |
| Task | 18744 | Create State Chart Diagram for ClubUML GUI | Ken Canaan | Done | Sprint 2 |
| Task | 18066 | Port Tab-Menu to ManagePolicy.JSP | Ken Canaan | Done | Sprint 2 |
| Task | 16487 | Port Tab-Menu to ManageContext.JSP and Create Header | Ken Canaan | Done | Sprint 2 |
| Task | 16486 | Port Tab-Menu to ManageProject.JSP and Create Header | Ken Canaan | Done | Sprint 2 |
| Task | 16482 | Port Tab-Menu into Promote.JSP | Ken Canaan | Done | Sprint 1 |
| Task | 16479 | Port Tab-Menu to Compare.JSP | Ken Canaan | Done | Sprint 1 |
| Task | 16472 | Port Tab-Menu in Display.JSP | Ken Canaan | Done | Sprint 1 |
| Story | 16319 | Project, Context, or Policy page Navigation | Ken Canaan | Done | Sprint 2 |
| Story | 15759 | Sprint 1 Navigation: Port Tab-Menu | Ken Canaan | Done | Sprint 1 |

## Tom Cashavelly

### Tom has the strengths of:

- System Design/Architecture

- Requirements

- GUI/Java Development

- REST Services Development

### Tasks

Currently right now there are 23 tasks within Jazz Hub that are allocated with Tom, with 7 of them still open. 4 of them are overlap with Ken’s effort which he was transferred the effort. The efforts within each tasks range from research, system design, development and documentation. A report of all tasks by Tom can be seen below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Id | Summary | Owned By | Status | Planned For |
| Defect | 16581 | After creating a project, it goes back to the Home page | Thomas Cashavelly | Resolved | Backlog |
| Story | 22374 | External Library Migration | Thomas Cashavelly | New | Sprint 4 |
| Task | 22371 | Modify Existing js, jsp, html files library references | Thomas Cashavelly | New | Sprint 4 |
| Task | 22370 | Migrate External Libraries to ClubUML | Thomas Cashavelly | New | Sprint 4 |
| Task | 22368 | Navigation Lessons Learned: Tom | Thomas Cashavelly | New |  |
| Story | 22367 | Navigation / System Design Lessons Learned | Thomas Cashavelly | New | Sprint 4 |
| Task | 22231 | Full System UML Design | Thomas Cashavelly | Done | Sprint 4 |
| Task | 21036 | GUI System Design Implementation | Thomas Cashavelly | Done |  |
| Task | 19771 | Create Index of ClubUML files and dependencies | Thomas Cashavelly | Done | Sprint 3 |
| Task | 19770 | Full System Design of all API and systems/subsystems | Thomas Cashavelly | Done | Sprint 4 |
| Task | 19402 | Create a Prototype for a new Modern Tab Navigation | Thomas Cashavelly | Done | Sprint 2 |
| Task | 18158 | Research to Re-Design/Architect GUI Framework | Thomas Cashavelly | Done | Sprint 2 |
| Task | 17318 | Develop GUI/Navigation Software Development Schedule | Thomas Cashavelly | Done | Sprint 2 |
| Task | 17003 | Add a header to ClassMerge.JSP | Thomas Cashavelly | Done | Sprint 1 |
| Task | 16576 | Create a Header in Manage Policy | Thomas Cashavelly | Done | Backlog |
| Task | 16489 | Sprint 1 Navigation Documentation | Thomas Cashavelly | Done | Sprint 1 |
| Task | 16488 | Port Tab-Menu to ManagePolicy.JSP | Thomas Cashavelly | Invalid | Backlog |
| Task | 16483 | Port Tab-Menu to ClassMerge.JSP | Thomas Cashavelly | Done | Sprint 1 |
| Story | 16477 | Navigation Modernization | Thomas Cashavelly | In Progress | Sprint 5 |
| Task | 16249 | Import Loading Bar | Thomas Cashavelly | New | Backlog |
| Task | 16111 | Research into the files need to migrate navigation pane | Thomas Cashavelly | Done | Sprint 1 |
| Story | 16110 | Re-Design and Architect GUI Framework | Thomas Cashavelly | Implemented | Sprint 3 |
| Task | 16109 | Research into JavaScript libraries to modernize navigation panels | Thomas Cashavelly | Done | Sprint 2 |

# Lessons Learned

* Have an understanding of SCRUM prior to the course
* Have an understanding of JazzHub prior to the course
* Have an understanding of the project prior to the course

## SCRUM

During the first 1-2 Sprints, there was an educational curve rather than diving into the implementation of the project. It took a few weeks for the class to get ramped up together to understand how SCRUM worked and a way forward, logistics wise. A lesson learn here is to recommend a small portion at the end of the Software Engineering course to review SCRUM to support the transition to the Software Project course.

## JazzHub

During the first 1-2 Sprints, there was an educational curve rather than diving into the implementation of the project. It took a few weeks for the class to get ramped up together to understand how JazzHub worked and a way forward. It took a few sprints to get our heads wrapped around the way it worked. It was also the first time epics, stories, tasks were introduced to us. A lesson learn here is to recommend a small portion at the end of the Software Engineering course to review JazzHub to support the transition to the Software Project course.

## Project

During the first 1-2 Sprints, there was an educational curve rather than diving into the implementation of the project. It took a few weeks for the class to get ramped up together to understand how the ClubUML system was. A lesson learned and was actually implemented, is to have a high level system diagram. The navigation team developed a diagram for future classes to read and understand the system. This will increase the ramp up time in the future for engineers to understand.

# Guiding Questions

What worked well with your tasking? What could be improved upon? How did you make decisions about how to accomplish the task?

Proper planning and breakdown of tasks in JazzHub helped the Navigation team complete sprints accordingly, and communicated via email for any internal team decision making.

However, we did not have the understanding and situational awareness of the other teams status. We did not always take advantage of the blackboard discussions that were not our own. We primarily used blackboard to update and communicate within the navigation team. We can improve to make sure we read and understand all of the discussions within the team. To make our big decisions, we brought our concerns and recommendation directly to the end of Sprint review. We let the entire team decide a way forward rather than making an internal team decision.

What are the processes (activities toward achieving goals) that you’re using? Are they effective? What other processes might work better? How do these compare to the processes learned in the Software Engineering class? What are the comparative advantages and disadvantages between your processes and the ones learned in class?

The Team followed an Agile Development approach to support the development of the Navigation/GUI Development for the semester. It is an iterative and incremental approach for development where requirements and solutions are evolved. Agile development promotes dynamic planning during development and the delivery of the software. As seen in the schedule, there are iterations and code deliveries after a level of effort. Each level of effort will consist of different functionality or a big milestone.

Each sprint lasts for a two week period and it broken up into 2 chunks with a scrum meeting halfway through the sprint. The GUI team members participated within each scheduled scrum meeting throughout the semester. A scrum meeting was a biweekly Tag up of the GUI team internally and also with the entire ClubUML team. The purpose of the scrum is to synchronize with all of the team members of each member’s current status. The GUI team members thoroughly communicated throughout the scrum, however it is beneficial for them to meet with the entire ClubUML team for an up to date status.

The Software Engineering class did not go into depth regarding Agile Development. The SE course briefly went over the overall PM approach to a software implementation. During the SE course, we were able to utilize the knowledge we learned for UML and state machine diagramming.

As the first time being in an Agile Development team environment, it has been very advantage to understand and track my progress during the year. It gives me the ability to update my own status and understand the teams status during the project.

Aspects of the Software Engineering course utilized during the project are the following:

* UML Diagrams: Sequence, State, Activity diagrams
* Project Management: Planning Activities, Laws of Project Management, Hierarchical Project Organization, Communication events (scheduled, unscheduled) and used both synchronous and asynchronous mechanisms
* System Decomposition
* Functional Model (System Decomposition) and Dynamic Model (Concurrency)
* Cohesion (high) and Coupling (low) of Subsystems
* Reuse (Started with Composition and finished with Inheritance)
* Unit Testing

What subject matter are you exploring? How does this subject matter inform the development of the application?

The Navigation Team has been providing the entire sustainment and development of the GUI/Navigation and also System Design. The overall effort has been collaborated efforts of research, reverse engineering, design, implementation and prototype modernization. The navigation team identified and prioritized the tasks level of effort during the course of the semester. During the development of the project, the team is becoming the SME for the GUI and system level design. This provides the ability for the entire team to understand the system at a high level and also to support integration/testing with the GUI. This has been seen during the bug team integration.

What technologies and communication techniques are you using? What has been particularly helpful, and what gaps are there?

There are several different technologies that are being used.

*For the GUI/Navigation:*

JavaScript

Java Server Pages

CSS

HTML

Java Servlet

*For System Design:*

UML – State/Class

*Communication Techniques:*

Blackboard, Blogs, and Email were used for daily communication among teams.

The most helpful tool has definitely been the Jazzhub tool. This provides the team the opportunity to re-connect and provide a status update to the entire team and keep track of progress of the entire semester. Instead of keep a list personally, a global tracker provides the ability to enter information about the project.

One gap here is that it took some time to learn the tool. In the beginning it felt like there was more time updating Jazzhub than the actual implementation effort for the effort.

What’s interesting about your work? How do you see it evolving in the future?

What is interesting is that we were able to dig down into the GUI Navigation code and being able to extend it to other pages within the same system. This was done by reverse engineering to understand how the subsystem worked. In addition, we also took the path to create a prototype of a new modern navigation bar. This provides the capability to evolve and modernize the overall look and feel of ClubUML. We are hoping that once our implementation is complete, the next class is able to finalize and implement the remaining capability of the navigation bar.

For the system design, we are aiming towards a 90-95% solution for the entire class and future class to understand. It took the entire class a couple of weeks to reverse engineer and understand the ClubUML system. The class had to take a bottom ups approach to understand the system. The use of a state diagram and high level dependency diagram will help future developers and customers to see the system.

# Questions for Discussions

1. At any point, did any other team feel like they are too separated outside of the class?

2. Did any group feel like they didn’t have any resources?

3. Did every group check everyone else’s blackboard discussions?

4. Did people learn any new technology in their respected area?

5. Can you say there is something you will definitely walk away from this class and use?