Thomas Travers
CS406
Advanced Software Engineering
Grantham University

Week 1 Lab

June 14, 2020

Agile Estimation
Apply Agile Estimation to approximate the effort required to satisfy the following use cases for the Study Abroad web application, described in appendix D, on page 667. Keep in mind that the estimation should include the time that it takes to design a solution that satisfies the use case, code, unit test the solution, perform integration, and user acceptance testing.

1. Place each use case on a story card or post-it note.

2. For each story card, or post-it note estimate the number of hours required for development. The number must be in the Fibonacci sequence.

3. List for each UC, your rationale for the effort estimation and the number of hours estimated.

4. Review your estimates, reconsider your rationale and revise estimates, if needed.

5. Sum the number of hours for a total level of effort.

6. Submit the list of UCs, your rationale, the number of hours estimated, and the total hours estimated.

For example:

UC01. Search for Programs | 2 hours | Rationale - The developer will reuse an already written a search algorithms for exchange programs.

UC02. Display Program Detail …

SAMS Project use cases

UC01. Search for Programs (Actor: Web User, System: SAMS) This use case allows a user to search overseas exchange programs using a variety of search criteria such as subject, semester, year, country and region. The system searches the database and displays a list of programs, each of which includes a link to display the detail of the program.

> ***UC01. Search for Programs | 8 hours | I chose 8 assuming writing algorithms from scratch. We should be able to accomplish this task using XML or JSON to communicate data to/from the database.***

UC02. Display Program Detail (Actor: Web User, System: SAMS) This use case retrieves and displays the detailed description of a program, selected by clicking the link of the program displayed by the Search for Programs use case, or by giving the program ID or program name.

> ***UC02. Display Program Detail | 3 hours | I chose 3 because all the data communication stuff should be completed in UC01 (functional dependency) leaving just the UI work here.***

UC03. Submit Online Application (Actor: Student, System: SAMS) "This use case allows a student to apply to an over- seas exchange program. The student fills in an application form and submits it. The system verifies the application, saves it in the database, and sets the status of the application to “submitted.” The system also sends e-mail to the two faculty members requested by the student to write recommendation letters, and the academic adviser to approve the course equivalency form.

> ***UC03. Submit Online Application | 21 hours | I chose 21 here because this use case has many tasks and functions that can be broken down.***

UC04. Login (Actor: Student, System: SAMS) This use case allows a registered student to login to the system.

> ***UC04. Login | 34 hours | I chose 34 because this task requires us to design a database for our project. The database must have carefully designed tables to prevent data corruption. It might be beneficial to also add LDAP as the authentication mechanism as this standard can be used for other applications on campus as well including campus email, student portal access, thus limiting account creations. If this system is already implemented, we can probably save a few hours by utilizing the existing LDAP system.***

UC05. Logout (Actor: Student, System: SAMS) This use case allows a student to logout from the system.

> ***UC05. Logout | 1 hour | I chose 1 because this is as simple as the front end terminating the connection and “forgetting” user credentials.***

UC06. Edit Online Application (Actor: Student, System: SAMS) This use case allows the student to edit an application that is not yet submitted.

> ***UC06. Edit Online Application | 3 hours | I chose 3. Assuming some previous tasks are dependencies, this is simply a form that auto-loads fields with existing values, allows edits, and submit or cancel changes.***

UC07. Check Application Status (Actor: Student, System: SAMS) This use case allows a student to check the status of an application, such as in-preparation, submitted, under-review, accepted, and rejected.

> ***UC07. Check Application Status | 2 hours | This is simply setting and retrieving a value in the database so I chose 2.***

UC08. Submit Recommendation (Actor: Faculty, System: SAMS) This use case lets a faculty member submit a recommendation on behalf of a student. The system saves the recommendation in the database.

> ***UC08. Submit Recommendation | 3 hours | This is simply a form for upload that records and registers in the database as well as links the record for the file to a pending application.***

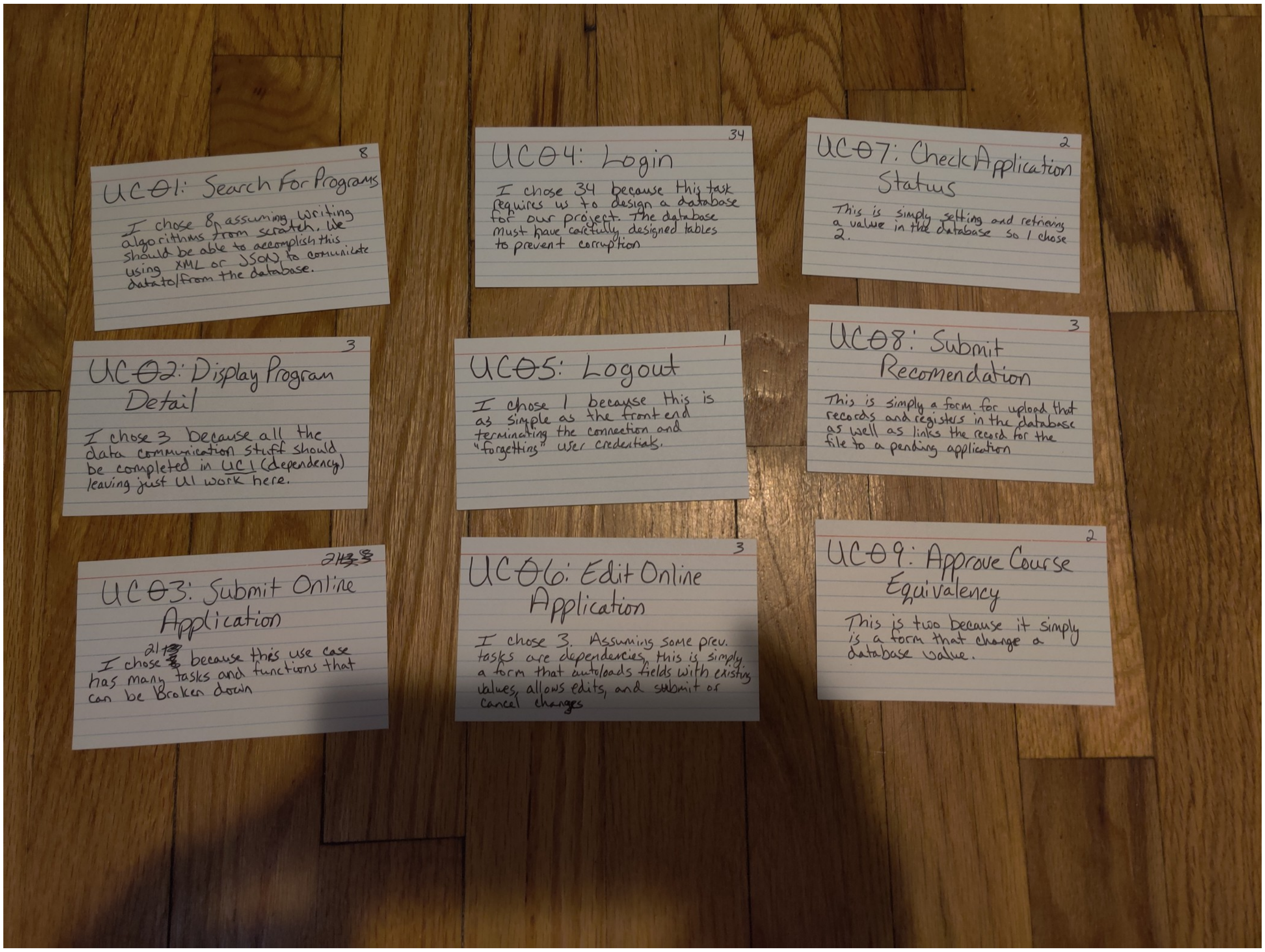
UC09. Approve Course Equivalency Form (Actor: Advisor, System: SAMS) This use case lets an academic adviser review and approve course equivalency forms submitted by students when they submit their online applications. Each form specifies the overseas courses that the student plans to use to substitute for the courses of the academic department."

> ***UC09. Approve Course Equivalency Form | 2 hours | This is 2 because it simply is a form that changes a database value.***

Use Case List	**Estimated Hours**
UC01: Search For Programs	*8*
UC02: Display Program Detail	*3*
UC03: Submit Online Application	*21*
UC04: Login	*34*
UC05: Logout	*1*
UC06: Edit Online Application	*3*
UC07: Check Application Status	*2*
UC08: Submit Recommendation	*3*
UC09: Approve Course Equivalency	*2*
Total Hours Estimated:	**77**

NOTE TO INSTRUCTOR: *Please bear in mind that I am still limited in experience so my agile method of common sense / experience for planning poker, or Fibonacci estimation may be a little bit inaccurate but may improve in time as I gain experience.*

Below is a picture of my story cards. I have written them into the assignment above but am including a picture to show that they were written on cards first.



8

UC01: Search For Programs

I chose 8 assuming writing algorithms from scratch. We should be able to accomplish this using XML or JSON to comunicate data to/from the database.

3

UC02: Display Program Detail

I chose 3 because all the data communication stuff should be completed in UC1 (dependency) leaving just UI work here.

21 ~~13~~ ~~8~~

UC03: Submit Online Application

I chose 21 ~~13~~ ~~8~~ because this use case has many tasks and functions that can be Broken down

34

UC04: Login

I chose 34 because this task requires us to design a database for our project. The database must have carefully designed tables to prevent corruption

1

UC05: Logout

I chose 1 because this is as simple as the front end terminating the connection and "forgetting" user credentials.

3

UC06: Edit Online Application

I chose 3. Assuming some prev. tasks are dependencies, this is simply a form that autoloads fields with existing values, allows edits, and submit or cancel changes

2

UC07: Check Application Status

This is simply setting and retrieving a value in the database so I chose 2.

3

UC08: Submit Recomendation

This is simply a form for upload that records and registers in the database as well as links the record for the file to a pending application

2

UC09: Approve Course Equivalency

This is two because it simply is a form that change a database value.