

Process Plan Document

Uplan

Richard Hartnett, Joe Menzie, Zach Matthews, Steve Lucey

1. Project Overview

The software product we are developing is a self-planning planner called UPlan. We are developing the product to require a user account for every user, to store their calendar data securely on a web server. When the user creates an account, they will fill in certain preferences such as sleep schedule to help the application schedule efficiently based on when the user goes to sleep. The next thing the user will have to do is add events. These events could be classes or other events going on in a specific week. Next the user will enter their assignments. These assignments will have fields such as: how long it will take to complete, the due date, and the priority of the assignment. UPlan's algorithm will take these assignments, find free time blocks from the list of events, and plan a study schedule based priority level and due date of the assignments. This study schedule will include which assignments to work on that day and for how long. The idea is, the user will use this generated schedule and roughly follow it to keep them on track for their day to day and long term assignments.

The user will be able to edit these events and assignments to alter the time they occur or other settings related to them. One very important aspect the user will need to know for the application to work correctly is at the end of each day or couple days, they will need to go back and update how much time they have left for each assignment. This needs to be done because a lot of time when someone starts an assignment they predict it will take a certain amount of hours but in reality it will take maybe half as long or twice as long. To factor in for this inaccuracy, the user will need to continuously update how far they have come and how much more they have to do.

2. Process Description

2.1 Project Lifecycle

We will be following the Iterative-and-Incremental lifecycle approach to develop our software. This approach will work well for our software because we have multiple definitive deadlines throughout our development where we will need working software for presentation. With this approach, we will create multiple working iterations of the software as we go, which guarantees us a working program once the first iteration is completed. This will also allow us to initiate meaningful testing of our program code early in the developmental process.

There are two planned iterations in our development process. We will essentially be doing the waterfall method for each of the iterations, following the sequence of activities shown in 2.2. We will be overlapping the iterations by starting activity 1 & 2 of the next iteration while we are finishing up with the previous iteration. The overlap will most likely occur at the end of implementation or during testing.

2.2 Process Activities

#	Activity	Description	Input	Output
1	Requirements & Analysis	Establish necessary features of software and time estimate for milestones	Knowing the basic goal of the particular iteration of the program	Process Plan & Software Requirements Specification (SRS)
2	Design	Create an outline for the	Process Plan & SRS	Design Document

		program highlighting how individual modules interact with each other to work coherently		
3	Implementation	Turn the design from step 2 into a working set of simple, understandable code	Design Document	Code and accompanying libraries/environment
4	Testing	Run test cases on the code to find any defects and correct them	Code and accompanying libraries/environment	Test plan & test cases
5	Maintenance	Continue to monitor program for additional bugs and add new features (i.e. the next iteration)	Working program from current iteration	Further refined program

3. Roles

3.1 Team member names

The team comprises of the following 4 persons:

- a. Richard Hartnett
- b. Steve Lucey
- c. Zachary Matthews
- d. Joseph Menzie

3.2 Roles Table

Roles	Description
Overall Project Manager	Delegate Tasks, Sets Schedules
Head of Testing	Manages Testing Goals/Schedule
Lead Developer	Command of Developing Decisions
Documentation Lead	Accountability for Required Documents
Interface Design Team	Creating/Coding Calendar Layout
Assignment/Event Team	Implementation of Parts of Calendar
Network Team	Account Creation/Management, Server Work
System Team	Application Creation, Clock and System Operations
Algorithm Team	Ordering Events, Prioritizing Events
Testing Team	Testing, Debugging, Calibrating Functions

3.3 Role Assignment Table

Roles	Team Members
Overall Project Manager	Richie
Head of Testing	Zach
Lead Developer	Steve
Documentation Lead	Zach
Interface Design	Joe
Assignment/Event Team	Joe and Zach
Network Team	Richie and Steve
System Team	Joe, Steve, Richie, Zach
Algorithm Team	Steve and Richie
Testing Team	Joe, Steve, Richie, Zach

4. Estimates

4.1 Effort Estimate

Following is the schedule, effort, code lines and expected defects for the 2 iterations:

Iteration Number	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Total Effort (person - hours)	Lines of Code	Defects
1	11/01/2016	11/18/2016	120	1500	8
2	11/18/2016	12/12/2016	197	5000	18

Total Effort for project : 337

Total Lines of code: 5000

Total Defects: 26

4.2 Schedule

We use the bottom up approach for estimation. In this we list the major modules and tasks and then estimate their effort and schedule. Task assignment to project members is also specified.

Version 0.1 Iteration:

#	Tasks	Estimated Effort (man - hrs)	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Person
1	Create Basic Client	20	11/1/16	11/5	Joe, Zach
2	Create Basic Database	20	11/6/16	11/11/16	Zach, Joe
3	Create Server	30	11/1/16	11/12/16	Richie, Steve
4	Create Server-Client Connections	20	11/4/16	11/12/16	Steve, Richie
5	Create Server-Database Connections	15	11/7/16	11/11/16	Steve, Richie
6	Text Based GUI	5	11/12/16	11/13/16	Richie
7	Application Testing	10	11/13/16	11/18/2016	All

Version 1.0 Iteration:

#	Tasks	Estimated Effort (man - hrs)	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Person
1	Converting App into Java Date Format	25	11/28/16	12/1/16	Steve, Richie
2	Design Scheduling Algorithm (based on due date priority)	60	12/2/16	12/10	Steve, Richie
3	Design GUI in Java Swing	60	11/26/16	12/10/16	Joe
4	Logo Creation	7	12/6/16	12/9/16	Zach
5	Link GUI to Client	20	12/9/16	12/11/16	Joe, Richie
6	Testing	25	12/5/16	12/14/16	Zach

Reflection:

Going into this project we highly underestimated the amount of time each of these tasks would take. Because of the fact that we did not know much about Servers, databases, client server interaction, and so many other things, doing simple tasks we estimated would be very quick actually ended up being double or triple the time. In terms of what we would do differently, first we should have split up doing parts of the code earlier. It was tough for us to understand how we were going to actually go forward and accomplish a task without the basic knowledge of certain processes and this made it difficult to split up tasks. Knowing what we know now we could split each task and probably complete this app in half the time. Another thing we would do differently is have one person designated to designing the gui earlier in the process. We pretty much knew what we wanted the app to look like early on so starting on that earlier would have eliminated a lot of confusion and stress towards the end. The last thing we should have done differently was connect to the web server earlier. We thought it would be as easy as plugging the IP and port into our code and it would work and essentially that is what we did but we had a lot of trouble setting up the actual server and our database kept getting malformed. This resulted in us having to run on localhost to demo our product.

As far as what we would do the same, our process worked very well. Richie and Steve were in charge of figuring out the client-server connection and designing the server while Zach and Joe went ahead and outlined what needed to be done for the database and the client. This allowed us to merge all the pieces together pretty effortlessly for the 0.1 version. Another thing we would not change is the people we partnered with on parts of the code. Joe and Zach were the most interested in the GUI and had a vision for what they wanted it to look like. They were able to learn java swing and implement the vision they had for the product. Richie and Steve worked very well together in designing the scheduling algorithm and incorporating all the elements needed on the backend to make sure the expected output was achieved for every input.