Chapter 05: Planning (Summary)

Planning remains a very necessary activity when Web apps are built. Everyone plans to some extent, but the limit and scope of planning and other activities related to planning normally varies from people to people involved in a web development project.

In real world we can see that the web engineers manage daily work of planning, monitoring, and mainly controlling technical tasks. Planning, monitoring, and coordinating the combined work of a Web engineering team is normally supervised by the project leader. Other stakeholders related to the project generally have little interest in the details of the planning activity, but they have great interest in the final outcome of the web engineering project.

Primary goal of planning a web application is to execute the given task properly which means the final outcome of web app must meet the requirements of the client within expected time. Without proper planning, it is almost impossible or extremely hard to maintain deadline of product submission to the stakeholders and meet their required services.

Like Abraham Lincoln once said: Give me 6 hours to chop down a tree, I'll spend the first 4 sharpening my axe. When we start planning we should start with a really basic wireframe. We should try to develop from a 'user-story' approach, so that it helps us in thinking about the basic design and requirements from the jump.

These are the questions we think about when laying out a new app:

1. **Who's going to be using the app?**

   When considering the 'who', we should think about the following questions:

* How techy are the people using this app?

* How much time will people generally want to spend on the app?

* In what environment will people most likely be using the app?

* In what mindset will most users be in?

2. **What are they going to be using it for?**

   For example: When we create a personal website, we considered the fact that it's mainly used for clients or companies to see the work of clients. So it needs to focus on their portfolio. Therefore, we should remove about 90% of the functionality. The client looking at our page is going to want to spend about 15 seconds on my page, so it's in the best interest of the website to remove all the barriers from seeing that portfolio.

3. **How can it be arranged in an intuitive way?**

   Once we know how and who will be using the page, how we you lay it out in a way that our target user will 'get it' in just a couple seconds.

   People using our app will most likely be younger, because it will be people looking to spontaneously go to a concert. So we can predict that my users will be anywhere from 20 to 40. More importantly, they're people who are very comfortable with using technology. We should imagine the mindset the average user is in will probably generally positive and in the mood to have fun. It will help us to imagine a very specific person using my app (or a very specific situation in which we would use the app.

So once we have an idea for a Web app, we should carefully plan it before we start to do any coding or design. This will be amazingly obvious to most of us, but it's a

point that cannot be overstated, whether we're creating a completely new app, or repurposing an existing application. This should be general purpose workflow, designed to help people get started; if you're an experienced enterprise developer then we'll probably have a workflow and practices of our own to apply to open Web app development, which is fine.

While developing a project in large scale we must do proper planning. To develop a meaningful project plan, we need clear understanding about the scope of the project. These scope of the project can be determined by context, information, objectives, functionality, construction & performance etc. We also need to focus on communication work products such as business motivation for the overall Web application, objective for the Web applications, types of users, information goal, applicative goal, increment's description, content objective, list of function, set of usage scenario etc. Besides these, we need to keep in mind that somethings still remain uncertain, even we manage communication activity very carefully. In planning activity, we can follow a certain way to implement the Web engineering project. So, we are to specify the actions and tasks required for our aiming project. To do so, forming a table may very helpful whereas the first section of the top row of the table for content objects. Then the functions can be listed next. This will make our work-progress visible to us effortlessly. After a brief discussion, our Web development team can select & distribute the tasks that will be used to model, construct, and deploy these content objects and functions.

Now our focus is to produce work product. We should keep the intermediate work products as minimum as possible. Three main criteria will guide our team.

These are:

(i) Avoiding excessive intermediate work product.

(ii) Constructing work-product which will be necessary for subsequent work product.

(iii) Useful information based work-product for the final increment.

As we have to maintain the quality of our project at our best, we need to follow the appropriate way to assess quality. By maintaining quality of the WebApp, our team can reduce the amount of rework that we must do. That efforts result in lower costs and, more importantly, improved time-to-market. So, we have explicitly

define a set of tasks that will lead us to our absolute goal. In order to ensure the quality of work products, our Web team will conduct pair walkthroughs.

We should make the app as simple as possible; focus on getting it to do one thing—or a few closely related things—well. If we have loads of different use cases we want to achieve, we might want to split them across different apps. Our app that may require a different experience on different platforms, so we may have to have separate lists.

Next, we should try to write a user-friendly summary of our app, which will entice people to download it and try it. If we can summarize it in a sentence, then our idea is probably a good fit for an app!

Once we have decided on your app's intent and target audience, it is always a good idea to start with paper sketches—we can try drawing out different rough screens to show what the app could look like, and what the workflow will be as the user uses our app. We'll probably want to do a separate set of sketches for desktop, mobile, tablet, TV if our functionality lists from above require it.

If we fail to manage changes properly, our WebApp may turn into a chaos. So, it is essential to have proper plan to manage changes efficiently. We will consider main three sources to decide whether we need to change or not.

These three sources are:

(i)Non-technical stakeholders.

(ii) End users.

(iii) Web Engineers suggestions.

Building a Web team is one of the most crucial factor in developing WebApp. When the whole team work together with good communication and interaction, the productivity increased dramatically compare to individual work. Our Web team should be self-organizing, fit to its environment, proper schedule etc. Our team must be a jelled team where every an each members share a common goal, a common culture, and overall a sense of litheness that makes the whole team unique.

Another thing that is usually considered obvious but often gets overlooked is testing. We should test as early and as often as possible, as fundamental mistakes discovered early on can save a lot of time and money further on in the project when a lot of development have been done. A general testing plan is as follows:

Once we have written your app's statement of functionality and target audience, we should share it with a number of colleagues, friends and family. Does it sound like a good idea from the outset, or does it sound ridiculous? Does it just need tweaking or moderate re scoping?

We should share our rough sketches for feedback as well. Is anything obvious missing? Would anything else add significantly to the experience?

Next, it is often a good idea to create a functional prototype that allows people to test key functionality and interactions. We should try to get a selection of real users outside the development team to test these interactions and see how well they fared. Even if we can't afford a professional user testing setup, then no matter—a selection of friends and family is often nearly as good, provided we become administer the right questions and tests.

As we work through developing the app, repeat the user testing procedure as many times as is sensible. Now we are working with the real app, test on as many varied browsers and devices as we can, starting with the primary support targets and working outwards. We must try to consider what is an acceptable experience on each browser and device, and don't just test normal usage—we should see how the app performs under stress, and with edge cases such as malicious data entry and really old browsers.

Near the end of the project, we must put a rigorous round of QA testing in to weed out any last minute evil bugs; the ones that always bite us on the neck when we least expect it.

As we know, a good team is one of the most fundamental asset to build successful WebApp product. So, we need to carefully manage our whole team. There are some research based tips to follow for building a successful team key of these are strong leadership, respect for individual talent, highly committed team members, having the mentality of sustainable momentum.

Risk management is a very crucial part of a WebApp project. It is a series of tasks that help Web team to understand and manages several encountered while developing the desired WebApp. In risk management part, we have to identify the risk properly.

All risks can be organized in three major categories: (i) people risks, (ii) product risks, (iii) process risks. And most importantly, team can identify risk by asking a powerful question like "What can go wrong?" during planning as well as developing session. Then creating a risk managing table, we can prioritize the probability of risk and evaluate the important risks likely to happen. This can be done wisely by contingency plan.

Developing a schedule is very important to ensure timely completion of the whole Web app. Our Web development team should make a comprehensive to-do list. Then dividing whole task into smaller sub-tasks and prioritize those sub-tasks. Finally, accomplishing one by one following schedule. We may follow macroscopic project scheduling for better result. Where every increment's detail is shortly listed and it is very easy to understand at a glance. In scheduling, prioritizing the subtasks is very crucial as there are interdependencies among several tasks. That's why we have to maintain the sequence of tasks strictly. A task network, also called an activity network which is a graphical representation of the task flow for a project will be very helpful in this manner.

Managing quality of a Web application after completing deployment for the end users may be overwhelming for the team sometimes. It is very tempting to discount quality while fixing problem encountered by end-users. Because fixing after deploying a WebApp is much more expensive and complex than its developing time. But we have to be very sincere about quality of our WebApp all time. To do so, we can follow some quality assurance mechanism. In addition to this, Web team can conduct a pair walkthrough as well as team walkthrough and note the quality problems & issues when encountered.

Managing changes in Web project is comparatively easy as we use to develop our final WebApp in several increment. So, in case stakeholder request any new features or other changes we may ask them to wait till the next increment. To assess the impact and criticality of changes, we can categorize each change into four classes. Then we can assess the change according to our classes.

Tracking the progress of overall WebApp is essential to take required action timely and more importantly meeting the deadline of final deployment. Tracking also motivates team about its progress and keep the working momentum. To

efficiently track the WebApp, we can keep our eyes on framework activities. Another way is to check how many user scenarios have been implemented.

Sometimes, we may need to outsource of our development work to reduce costs and developing time. Because, our stakeholder may want such feature or specification that's not very well known to our team. So, we can outsource that particular part of the whole project.

So if we summarize the entire idea what we get?

Firstly we should create a list of tasks, an idea for our app and the type of user we are targeting, and then write a goal statement.

Define our app's purpose and the most important user in one sentence if possible. Example: A wish list creation tool for people who never do impulse shopping. It's possible that we cannot include all the tasks on our list in your goal statement. That is OK, because awesome apps — especially those for mobile — do one thing well. If our app is trying to do too many things, then we should think about splitting up the functionality over multiple apps.

Now we've identified our main use case, target users, and key features. Our main scenario should also consider the user environment in which our app is used.

There is a proverb that, "One minute in proper planning saves ten minutes in execution". So, we should efficiently prepare our planning activity for developing an awesome project which will be both cost efficient and quality WebApp.

Md. Towhidul Islam

Roll: 170132

Level 3, Term 2

Session: 2016-17

CSE 9 batch, PUST