

Assignment 1

1. What are four essential difficulties of software systems discussed in Fred Brooks's paper? Explain each using your own words.

=====

Essential difficulties are the ones which are very hard to target and usually occurs during software design, conceptualization, understanding and testing the conceptual construct. The four difficulties discussed in Fred Brooke's paper are below:

1. **Complexity** – Software design are far more complex due to lot of inter dependency among all the entities involved. Due to large number of states it's a huge challenge to understand, describe and test them. Scaling up of software entity necessarily means non-linear scaling of components resulting in more complexity. Complex functions notches up difficulty level one up. This also results in communication gap between team members and management issues arises. The overview of the software take a toss due to undesired complexity and requires lot of effort on all fronts thus increasing turnover.
2. **Conformity**- Software must conform to external constraints like hardware, regulations, legacy systems support. Like the launch of any new browser version in the market must conform to its legacy ones. New products just get confirmed but others do need to conform. This results in more arbitrary complexity as between human and software, everyone expects software to change & conform instead of human systems or rest of the system.
3. **Changeability** – With constant technology evolving and changing, it's a challenge for the software to be able to be functional and kept open to quick flexible changes. The changes are mainly because of physical machine changes, OS changes and users behavior. Users deviates from predetermined manner to use the software and get foreseen in software design. Even a small change in software ripples throughout the system and induces high cost. The maintenance of software is more expensive than the cost of developing it and constant changes in the rest of the system makes it as an inherent difficulty.
4. **Invisibility** - Software design can't be visualized as designs for physical structures and everyone have their own perception to look at software design. This inhibits the understanding between different team members as what one thought is not what other perceive. This also deprive one of one's capability to conceptualize it.

2. Pick one software method or tool that you used before and **Specifically** explain whether or not you think this method or tool is "promising attack" on the essential difficulties mentioned above.

=====

Assignment 1

With my experience in last years for a product company, we needed a change in our aggregation technology. Earlier we use to aggregate data using Microsoft Internet Explorer browser. Later on with rapid changes in stabilized version of IE we needed a better platform which can withstand the market changes.

The new platform utilized Selenium Webdriver and definitely focused on the above mentioned essential features. These are explained below:

1. **Changeability:** Since Selenium webdriver is a testing framework for all web application w.r.t to all browsers, it act as base layer for all browsers. Hence even if the browser changes, it is very easy to integrate the new high speed browser as the base remains the same. All the software modules coded for selenium webdriver remains same and only one module need to be changed for new browser. The system was designed into manageable modules and hence effectively address this difficulty.
2. **Conformity:** The software product written on top of this tool will conform to all legacy browsers unless and until selenium webdriver goes out of market. Also the API for selenium webdriver follows all are standards defined and hence will not induce much complexity. The new functionality can keep on adding like any Object oriented programming language.
3. **Complexity:** This tool also reduced complexity in testing phase as the testers don't need to test different websites for data aggregation w.r.t. new browser as the Selenium webdriver takes care of it. The testers and developers just need to concentrate on functional requirements rather than browser adaptability and external factors.
4. **Invisibility:** In my view this helped in simplifying the design process by giving much clear view of the layers involved in new platform. It definitely don't create any doubt of multiple views of the same system. But I must admit it partially addresses this and still there is scope of this essential difficulty.

Also In the last I want to mention that in my view No single tool or software can address all of them above. We need a mix of tools and effective process to address this essential difficulty in today modern world. For example If we use Selenium Webdriver with Perforce Change management tool ,Bugzilla , pair programming and Agile development process will definitely decrease essential difficulties.

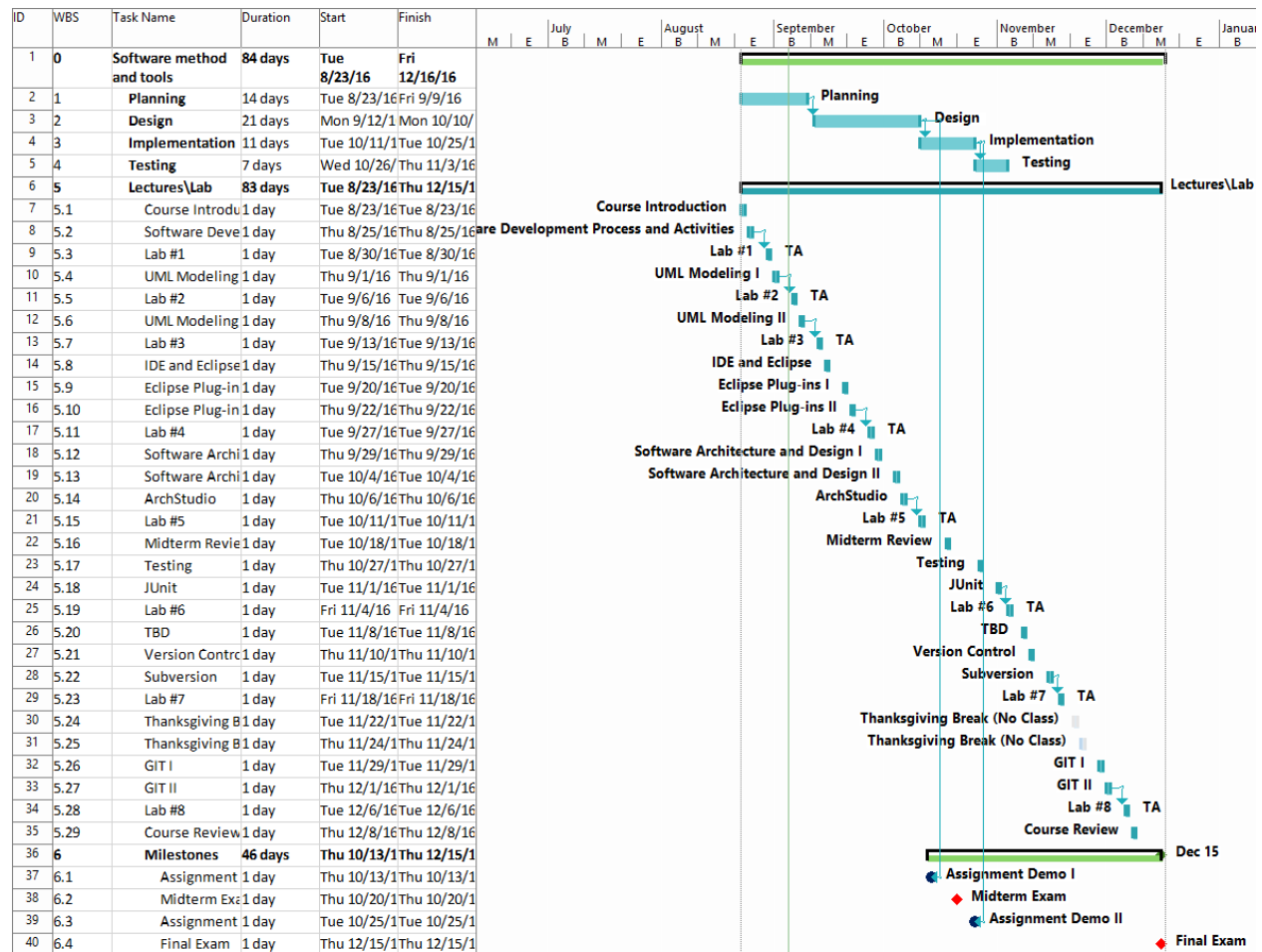
3. **Make a class schedule for this course using Microsoft Project 2013. Your schedule chart must include the following elements:**
 - A Project Summary Task named "Software Methods and Tools"
 - Summaries and Tasks (e.g. Planning, Design, Implementation, Testing)
 - Recurring Tasks (e.g. Lectures/Labs)
 - Milestones (e.g. Assignments, Exams)

Assignment 1

- Relationships between the above elements (e.g. the Planning task generates Assignment 1)

Below are the snapshots created with the help of Microsoft Project :

GANTT Chart:



Assignment 1

TASK TABLE:

ID	WBS	Task Name	Duration	Start	Finish	Predecessors	Resource Names	
1	0	Software method and tools	84 days	Tue 8/23/16	Fri 12/16/16		Yongjie Chang, Rashmi, TA	
2	1	Planning	14 days	Tue 8/23/16	Fri 9/9/16			Planning
3	2	Design	21 days	Mon 9/12/16	Mon 10/10/16			Design
4	3	Implementation	11 days	Tue 10/11/16	Tue 10/25/16			Implementation
5	4	Testing	7 days	Wed 10/26/16	Thu 11/3/16			Testing
6	5	Lectures\Lab	83 days	Tue 8/23/16	Thu 12/15/16		Yongjie Chang, TA	
7	5.1	Course Introduction	1 day	Tue 8/23/16	Tue 8/23/16			
8	5.2	Software Development	1 day	Thu 8/25/16	Thu 8/25/16			
9	5.3	Lab #1	1 day	Tue 8/30/16	Tue 8/30/16	8FS+2 days	TA	
10	5.4	UML Modeling	1 day	Thu 9/1/16	Thu 9/1/16			
11	5.5	Lab #2	1 day	Tue 9/6/16	Tue 9/6/16	10FS+2 days	TA	
12	5.6	UML Modeling	1 day	Thu 9/8/16	Thu 9/8/16			
13	5.7	Lab #3	1 day	Tue 9/13/16	Tue 9/13/16	12FS+2 days	TA	
14	5.8	IDE and Eclipse	1 day	Thu 9/15/16	Thu 9/15/16			
15	5.9	Eclipse Plug-in	1 day	Tue 9/20/16	Tue 9/20/16			
16	5.10	Eclipse Plug-in	1 day	Thu 9/22/16	Thu 9/22/16			
17	5.11	Lab #4	1 day	Tue 9/27/16	Tue 9/27/16	16FS+2 days	TA	
18	5.12	Software Architecture and Design I	1 day	Thu 9/29/16	Thu 9/29/16			
19	5.13	Software Architecture and Design II	1 day	Tue 10/4/16	Tue 10/4/16			
20	5.14	ArchStudio	1 day	Thu 10/6/16	Thu 10/6/16			
21	5.15	Lab #5	1 day	Tue 10/11/16	Tue 10/11/16	120FS+2 days	TA	
22	5.16	Midterm Review	1 day	Tue 10/18/16	Tue 10/18/16			
23	5.17	Testing	1 day	Thu 10/27/16	Thu 10/27/16			
24	5.18	JUnit	1 day	Tue 11/1/16	Tue 11/1/16			
25	5.19	Lab #6	1 day	Fri 11/4/16	Fri 11/4/16	24FS+2 days	TA	
26	5.20	TBD	1 day	Tue 11/8/16	Tue 11/8/16			
27	5.21	Version Control	1 day	Thu 11/10/16	Thu 11/10/16			
28	5.22	Subversion	1 day	Tue 11/15/16	Tue 11/15/16			
29	5.23	Lab #7	1 day	Fri 11/18/16	Fri 11/18/16	28FS+2 days	TA	
30	5.24	Thanksgiving Break	1 day	Tue 11/22/16	Tue 11/22/16			
31	5.25	Thanksgiving Break	1 day	Thu 11/24/16	Thu 11/24/16			
32	5.26	GIT I	1 day	Tue 11/29/16	Tue 11/29/16			
33	5.27	GIT II	1 day	Thu 12/1/16	Thu 12/1/16			
34	5.28	Lab #8	1 day	Tue 12/6/16	Tue 12/6/16	33FS+2 days	TA	
35	5.29	Course Review	1 day	Thu 12/8/16	Thu 12/8/16			
36	6	Milestones	46 days	Thu 10/13/16	Thu 12/15/16		Rashmi	
37	6.1	Assignment Demo I	1 day	Thu 10/13/16	Thu 10/13/16	FF+3 days		
38	6.2	Midterm Exam	1 day	Thu 10/20/16	Thu 10/20/16			
39	6.3	Assignment Demo II	1 day	Tue 10/25/16	Tue 10/25/16	FF		
40	6.4	Final Exam	1 day	Thu 12/15/16	Thu 12/15/16			