Ankita Wankhede
September, 2016

Assignment 1

No Silver Bullet: Essence and Accidents of Software Engineering

1. (1) What are four essential difficulties of software systems discussed in Fred Brook's paper? explain each using your own words.

Answer. According to Fred Brook's there are four essential difficulties of software system:-

1.1.Complexity

1.2.Conformity

1.3.Changeability

1.4.Invisibility

1. Complexity:-

The Author here tries to explain how software design is complex, and why it is essential that we understand the concept of entities to the extent, that entities and the relationships between them can be defined exactly, though this is hard it is essential if we want to translate this concepts into code. unlike physical objects, no two parts of a software construct are alike.

A software has numerous states which makes it difficult to design develop and test for example a Mechanical switch systems are easy to build, a switch which goes on and off is simple to develop and test, but when we try to perform a same function with software say a start/ stop, there are going to be n number of possibilities that will have to be addresses before the software performs Start/Stop. Likewise scaling up a software entity would mean scaling up in entities and complexity.

Thus Complexity is one of the essential property of software which also brings up the problem associated with it: difficulty of communication of design among team members; difficulty in enumerating all possible states of the program (which brings unreliability); makes programs hard

to use; makes programs hard to change without inducing unwanted side-effects. The complexity makes overview hard (seeing and tying up loose ends), and the tremendous learning burden means turnover is costly.

2. Conformity:

Software must conform to external constraint, though conformity may limit/ restrict and add complexity to the design it still has to abide by certain standards, hardware restrictions or regulations. Since most of the complexity of interacting with human systems is arbitrary. it is the software that must conform as its has come to the scene the most recently and can be easily changeable.

3. Changeability:

Software entity is constantly changing, yesterdays mainframe systems have become Oracle, Sybase systems moved to SQL, being on Cloud is the next big thing every industry wants to be able to provide its customer, why bring them up, because the software change is easy. best example are our mobile platforms at one time Java, Symbian were one of the most sought platforms for mobile development but with constant changes to the industry ease of availability and development, today most of the mobile platform are Android and IOS based.

Softwares are inherently flexible to make change to compared physical product, in short software products are mixed with lot of arbitrary factors like hardware, users, laws and regulations. this factors change continually, and in result cause changes to the software product.

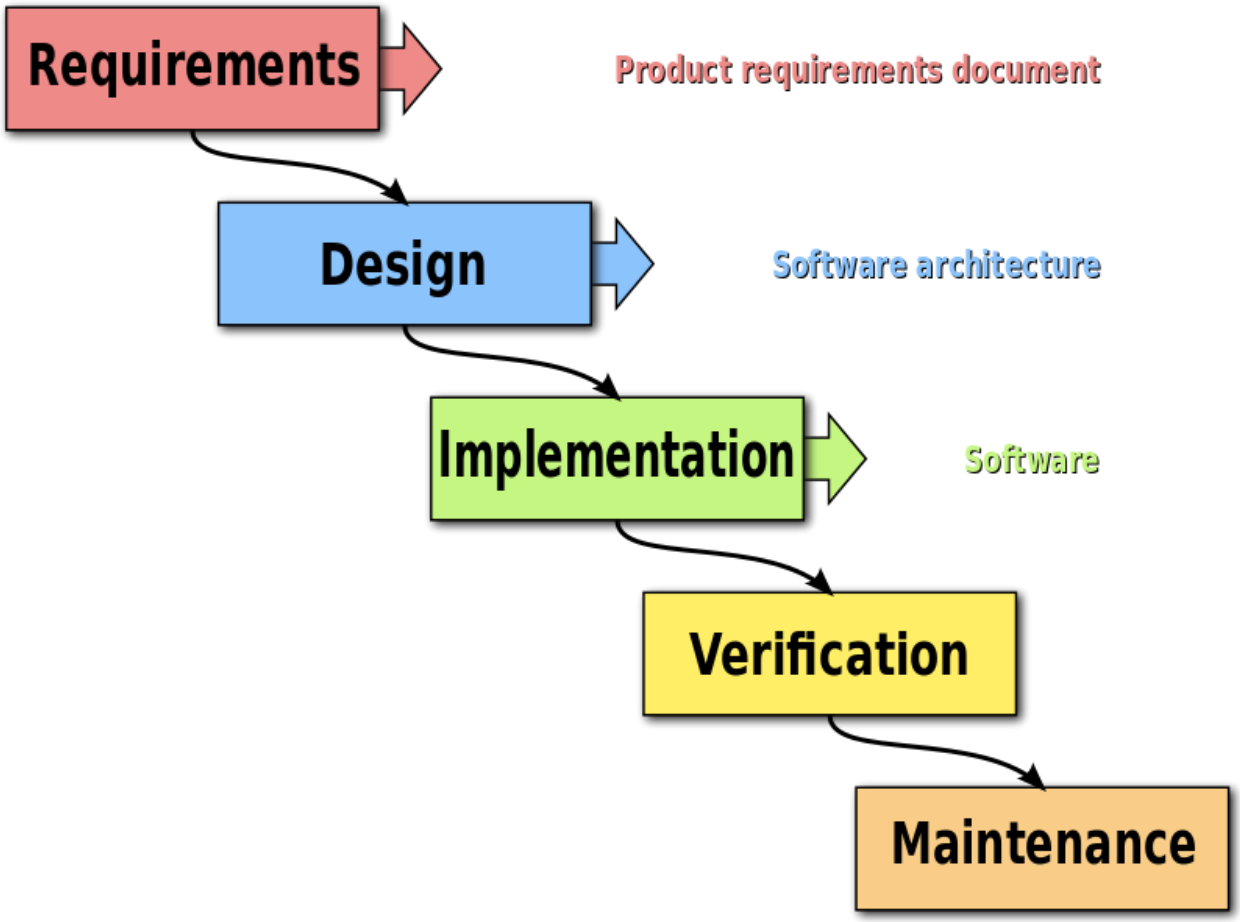
4. Invisibility:

Software is logic and lines of instructions for machines to function and is invisible in nature. It is also very difficult to visualize as there are many ways to look at the same design. the reality of software is not embedded in space and so it cannot have a figurative or a diagrammatic representation, the more we try to represent them in structured way we find there are multiple ways to represent the the same entity. We use several methods to explain the software like Use case, UMLs, architecture, flow diagrams etc. though they try to simplify and explain how the software interacts how the workflow is established the actual software in itself is still invisible for end users.

1. (2) Pick one software method or tool that you used before and specifically explain whether or not you think this method or tool is a “promising attach” on the essential difficulties mentioned above.

Answer. Waterfall Model is one of the most well known software development methodology used in the IT industry, While working for Huawei Technologies I was exposed exclusive to this development model, and I want to talk more about this methodology and the essential difficulties I faced while working on projects under waterfall model.

The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.



Complexity: Though at one time the Waterfall model was one of the most used development methodologies it slowly has lost its fame as preferred method due to the complexity of issues faced. One of the key drawback was that the clients don't realize exactly what their requirements are until they see a working software which leads to redesign, redevelopment and increase in cost. also this model often adds complexity for developer since they are unaware of the future difficulties when designing the features. however the model does not account for these complexities. The method is not flexible and the turn around time are too large for designers to handle complex code development.

For example if a defect is introduced while capturing requirements and not found till User Acceptance Testing this creates a bug for which the designers never accounted for and unnecessary code complexity has been introduced because of this, fixing which may introduce new bugs in the code.

This software method may work for large projects, however with present changing markets its hard for client to wait for long to introduce new products and features into the market, as there is always a chance that there competitors are also building something new.

2. Conformity: Waterfall SDLC method has always been in line with essence of conformity, the long period of analysis helps, keep the designs in line with the Laws and regulations along with being able to analyze the system requirements for which the design needs to be created.

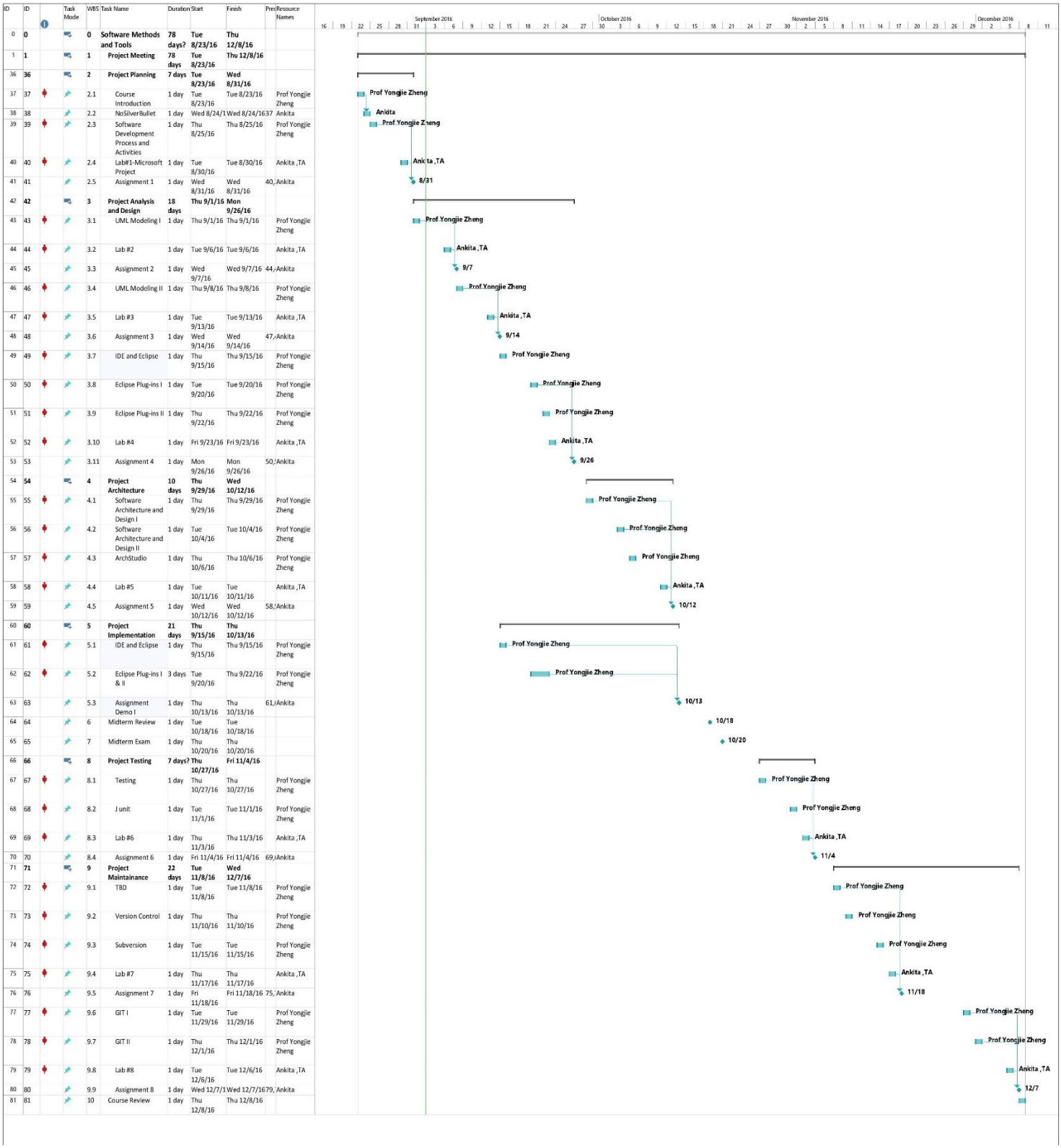
3. Changeability: Waterfall model is capable of handling the changeability essence due to its long span of operation, but at the same time the method is not cost effective if the change in requirements come at a later stage of development, and in my experience the requirements keep changing though out the SDLC phase. a software goes thorough this phase for multiple reasons like, unforeseen circumstances or requirements, change in technology and defect requiring to change the design approach, though the waterfall model accounts for this essence of difficulty but is not the most cost effective method for client.

4. Invisibility : with a huge amount of time spent in requirement phase, this software method do offer the most near structure documentation of the product/ feature with System requirements documents, Use cases, UML documentation however still the design logic stays unvisualized often the requirements get changes when the software is delivered and very often is the case that the requirements have changed or the client needs new feature which they did not realize at the start. though this software method makes the most attempt to bring the design as near to visualization as possible, it often leaves lot of areas unvisualized like what solutions logic needs to be implemented, what approach is the best and least problematic approach

Answer 2:

Please find my below Microsoft Project elements and elements table:-

Elements	Name of the Elements
Project Summary	Software Methods and Tools
Summaries	Project Meeting, Project Planning, Project Analysis & Design, Project Architecture, Project Implementation, Project Testing, Project Maintenance
Recurring Tasks	Lectures should have held on every Thursday Lab should have held on every Tuesday
Tasks	Course Introduction, Software Development Process and Activities, Microsoft Project, UML Modeling I&II, Eclipse Plugin-ins I&II, SAD-I&II, Arch studio, IDE & Eclipse, Testing, JUnit, TBD, Version control, subversion, GIT I&II
Milestones	Assignments 1 to 8, Midterm Review, Midterm Exam
Relation	After completion of each task (Labs &Lectures) you have to submit an assignment. Suppose if we have completed lab and lectures of Project Testing we have to submit Assignment 6 on Monday.
Resources	Prof Y Zheng – He will cover all the Lectures Ankita & TA-They will take part in Labs Ankita – She will submit Assignment at the end of each Lectures &Labs.



ID	ID	Task Mode	WBS	Task Name	Duration	Start	Finish	Pred	Resource Names
0	**0**		**0**	**Software Methods and Tools**	**78 days?**	**Tue 8/23/16**	**Thu 12/8/16**		
1	**1**		**1**	**Project Meeting**	**78 days**	**Tue 8/23/16**	**Thu 12/8/16**		
36	**36**		**2**	**Project Planning**	**7 days**	**Tue 8/23/16**	**Wed 8/31/16**		
37	37		2.1	Course Introduction	1 day	Tue 8/23/16	Tue 8/23/16		Prof Yongjie Zheng
38	38		2.2	NoSilverBullet	1 day	Wed 8/24/16	Wed 8/24/16	37	Ankita
39	39		2.3	Software Development Process and Activities	1 day	Thu 8/25/16	Thu 8/25/16		Prof Yongjie Zheng
40	40		2.4	Lab#1-Microsoft Project	1 day	Tue 8/30/16	Tue 8/30/16		Ankita ,TA
41	41		2.5	Assignment 1	1 day	Wed 8/31/16	Wed 8/31/16	40,	Ankita
42	**42**		**3**	**Project Analysis and Design**	**18 days**	**Thu 9/1/16**	**Mon 9/26/16**		
43	43		3.1	UML Modeling I	1 day	Thu 9/1/16	Thu 9/1/16		Prof Yongjie Zheng
44	44		3.2	Lab #2	1 day	Tue 9/6/16	Tue 9/6/16		Ankita ,TA
45	45		3.3	Assignment 2	1 day	Wed 9/7/16	Wed 9/7/16	44,	Ankita
46	46		3.4	UML Modeling II	1 day	Thu 9/8/16	Thu 9/8/16		Prof Yongjie Zheng
47	47		3.5	Lab #3	1 day	Tue 9/13/16	Tue 9/13/16		Ankita ,TA
48	48		3.6	Assignment 3	1 day	Wed 9/14/16	Wed 9/14/16	47,	Ankita
49	49		3.7	IDE and Eclipse	1 day	Thu 9/15/16	Thu 9/15/16		Prof Yongjie Zheng
50	50		3.8	Eclipse Plug-ins I	1 day	Tue 9/20/16	Tue 9/20/16		Prof Yongjie Zheng
51	51		3.9	Eclipse Plug-ins II	1 day	Thu 9/22/16	Thu 9/22/16		Prof Yongjie Zheng
52	52		3.10	Lab #4	1 day	Fri 9/23/16	Fri 9/23/16		Ankita ,TA
53	53		3.11	Assignment 4	1 day	Mon 9/26/16	Mon 9/26/16	50,	Ankita
54	**54**		**4**	**Project Architecture**	**10 days**	**Thu 9/29/16**	**Wed 10/12/16**		
55	55		4.1	Software Architecture and Design I	1 day	Thu 9/29/16	Thu 9/29/16		Prof Yongjie Zheng
56	56		4.2	Software Architecture and Design II	1 day	Tue 10/4/16	Tue 10/4/16		Prof Yongjie Zheng
57	57		4.3	ArchStudio	1 day	Thu 10/6/16	Thu 10/6/16		Prof Yongjie Zheng
58	58		4.4	Lab #5	1 day	Tue 10/11/16	Tue 10/11/16		Ankita ,TA
59	59		4.5	Assignment 5	1 day	Wed 10/12/16	Wed 10/12/16	58,	Ankita
60	**60**		**5**	**Project Implementation**	**21 days**	**Thu 9/15/16**	**Thu 10/13/16**		
61	61		5.1	IDE and Eclipse	1 day	Thu 9/15/16	Thu 9/15/16		Prof Yongjie Zheng
62	62		5.2	Eclipse Plug-ins I & II	3 days	Tue 9/20/16	Thu 9/22/16		Prof Yongjie Zheng
63	63		5.3	Assignment Demo I	1 day	Thu 10/13/16	Thu 10/13/16	61,	Ankita
64	64		6	Midterm Review	1 day	Tue 10/18/16	Tue 10/18/16		
65	65		7	Midterm Exam	1 day	Thu 10/20/16	Thu 10/20/16		
66	**66**		**8**	**Project Testing**	**7 days?**	**Thu 10/27/16**	**Fri 11/4/16**		
67	67		8.1	Testing	1 day	Thu 10/27/16	Thu 10/27/16		Prof Yongjie Zheng
68	68		8.2	J unit	1 day	Tue 11/1/16	Tue 11/1/16		Prof Yongjie Zheng
69	69		8.3	Lab #6	1 day	Thu 11/3/16	Thu 11/3/16		Ankita ,TA
70	70		8.4	Assignment 6	1 day	Fri 11/4/16	Fri 11/4/16	69,	Ankita
71	**71**		**9**	**Project Maintainance**	**22 days**	**Tue 11/8/16**	**Wed 12/7/16**		
72	72		9.1	TBD	1 day	Tue 11/8/16	Tue 11/8/16		Prof Yongjie Zheng
73	73		9.2	Version Control	1 day	Thu 11/10/16	Thu 11/10/16		Prof Yongjie Zheng
74	74		9.3	Subversion	1 day	Tue 11/15/16	Tue 11/15/16		Prof Yongjie Zheng
75	75		9.4	Lab #7	1 day	Thu 11/17/16	Thu 11/17/16		Ankita ,TA
76	76		9.5	Assignment 7	1 day	Fri 11/18/16	Fri 11/18/16	75,	Ankita
77	77		9.6	GIT I	1 day	Tue 11/29/16	Tue 11/29/16		Prof Yongjie Zheng
78	78		9.7	GIT II	1 day	Thu 12/1/16	Thu 12/1/16		Prof Yongjie Zheng
79	79		9.8	Lab #8	1 day	Tue 12/6/16	Tue 12/6/16		Ankita ,TA
80	80		9.9	Assignment 8	1 day	Wed 12/7/16	Wed 12/7/16	79,	Ankita
81	81		10	Course Review	1 day	Thu 12/8/16	Thu 12/8/16		