Phanideep Salapaka

ID:16198748

Software Methods and Tools

Assignment1

1)

1) Four essential difficulties of software systems discussed in Fred Brooks's paper are

- Ø Complexity
- Ø Conformity
- Ø Changeability
- Ø Invisibility

Complexity:

Complexity in software entities is directly related to their size because no two parts are similar, which is not in the case with other entities like computers, automobiles where there can be repeated elements combined to form large systems. The complexity is more in digital computers which have large number of states which makes describing and testing them hard.

The complexity in software systems is not an accidental property but is an essential one. This can be explained by studies in fields like mathematics and physics where complex phenomena are simplified because for these fields complexities ignored here are not essential properties. This similarity doesn't work in cases where complexity is an essence.

Scaling-up software systems is not repeating same elements in large size but it is increasing number of different elements. These systems communicate with each other in nonlinear fashion which makes systems more complex. Thus complexity leads to communication faults which leads to product flaws, increased costs, difficulty in enumerating things which finally leads to unreliability.

Apart from these technical difficulties there are management difficulties as well that come from complexity. These all makes financial burden a disaster.

Conformity:

The most complexities in the software systems are arbitrary which is forced by human interactions and systems should confirm many non standard modules and user interfaces. The systems should confirm these not only because of necessity but also because it is designed by different people. Software systems should confirm because it expected as most conformable. In many cases complexity comes from confirmation to modules developed by other users. These complexities cannot be reduced merely by redesigning the software alone.

-
-
-

Changeability:

Software system is constantly subjected to changes. Changeability is also there in all fields like buildings, cars, computers etc. But changes in these things are infrequent after manufacturing. The changes appear in their later models, automobiles are recalled infrequently and buildings are expensive to remodel. All these are less frequent when compared to changes applied on fielded software.

The changeability is common in software because of many reasons. First, it is a function of a system which has more pressure of change. Second, the software can be changed easily. Third, all successful projects tend to change because people found the project helpful and they want extended features in the particular project. Fourth, software products are used in all generally used applications which change continually and these changes force changes to software product.

Invisibility:

Software is invisible and unvisualizable. Some things like floor plans of buildings, scale drawings of mechanical parts help to identify problems and optimizations of space. But in case of software its hard to diagram software visually. We find that one diagram may consist of many overlapping graphs rather than just one. These several graphs represent flow of control, flow of data, patterns of dependency etc. These graphs are not hierarchical and one of the ways of getting control of these structures is to make these graphs hierarchical.

Even after there is progress in simplifying these structures they remain unvisualizable which deprives the engineer from using the brain's powerful visual skills. This will affect not only process of design within one mind but also affects communication among minds.

2) The software tool I have used before is Eclipse IDE tool.

Through paper Brook lists "promising attacks" on the essential difficulties as the following:

- Ø Buy versus build
- Ø Requirement refinement and rapid prototyping
- Ø Great designers

Eclipse IDE tool is free and open source software widely used in developing java based applications. This tool satisfies the requirements to have promising attack on essential difficulties.

Buy Vs Build: Eclipse IDE tool is once built and it is available for particular cost. It is not re-build from scratch every time. So Eclipse IDE complies with Buy Vs Build promising attack

Requirement refinement and rapid prototyping: This tells to first develop software with essential features as early as possible and refine the product with updates. Eclipse IDE was first released in 2001 with some functionality and updates are being released from time to time to get advanced features. Advanced features include different plug-ins, feasibility in coding, addition of new SDKs, extension for other programming languages etc. So it complies with Requirement refinement and rapid prototyping attack.

Great Designers: Great designers are considered as more effective than normal designers. Eclipse IDE is designed by some famous organizations like IBM, Merant, RedHat, QNX Software Systems. As Eclipse IDE tool is designed by giant companies it complies with great designers promising attack.

2. Class schedule for SMT course using Microsoft Project 2013.

ID	Task Mode	WBS	Task Name	Duration	Start	Finish	Predecesso	Resource Names
0		**0**	**Project1**	**78 days**	**Tue 9/1/15**	**Thu 12/17/15**		
1		1	Software Methods and Tools	74 days	Mon 9/7/15	Thu 12/17/15		
2		**2**	**Planning**	**17 days**	**Mon 9/7/15**	**Tue 9/29/15**		
3		**2.1**	**Lecture**	**11 days**	**Thu 9/10/15**	**Thu 9/24/15**		**Zheng**
4		2.1.1	Planning introductior	1 day	Thu 9/10/15	Thu 9/10/15		
5		2.1.2	Requirements collection	1 day	Thu 9/17/15	Thu 9/17/15		
6		2.1.3	Analysis	1 day	Thu 9/24/15	Thu 9/24/15		
7		**2.2**	**Lab**	**6 days**	**Tue 9/8/15**	**Tue 9/15/15**		**Phanideep**
8		2.2.1	Project 2013	1 day	Tue 9/8/15	Tue 9/8/15		
9		2.2.2	UML modelling	1 day	Tue 9/15/15	Tue 9/15/15		
10		**3**	**Assignments set 1**	**11 days**	**Tue 9/1/15**	**Tue 9/15/15**	**2**	**Phanideep**
11		3.1	Assignment 1	5 days	Tue 9/1/15	Mon 9/7/15		
12		3.2	Assignment 2	5 days	Tue 9/8/15	Mon 9/14/15		
13		**4**	**Design**	**12 days**	**Wed 9/30/15**	**Thu 10/15/15**		
14		**4.1**	**Lecture**	**11 days**	**Thu 10/1/15**	**Thu 10/15/15**		**Zheng**
15		4.1.1	Architecture patterns	1 day	Thu 10/1/15	Thu 10/1/15		
16		4.1.2	Styles	1 day	Thu 10/8/15	Thu 10/8/15		
17		4.1.3	Functional design	1 day	Thu 10/15/15	Thu 10/15/15		
18		**4.2**	**Lab**	**8 days**	**Tue 10/6/15**	**Thu 10/15/15**		**Phanideep**
19		4.2.1	Archstudio	1 day	Tue 10/6/15	Tue 10/6/15		
20		**5**	**Assignments set 2**	**7 days**	**Tue 10/6/15**	**Wed 10/14/15**	**4**	**Phanideep**
21		5.1	Assignment 3	5 days	Tue 10/6/15	Mon 10/12/15		
22		5.2	Assignment 4	2 days	Tue 10/13/15	Wed 10/14/15		
23		**6**	**Implementation**	**20 days**	**Fri 10/16/15**	**Thu 11/12/15**		
24		**6.1**	**Lecture**	**11 days**	**Thu 10/22/15**	**Thu 11/5/15**		**Zheng**
25		6.1.1	Integrated Development environment	1 day	Thu 10/22/15	Thu 10/22/15		
26		6.1.2	Coding techniques	1 day	Thu 10/29/15	Thu 10/29/15		
27		6.1.3	Java basics	1 day	Thu 11/5/15	Thu 11/5/15		
28		**6.2**	**Lab**	**11 days**	**Tue 10/20/15**	**Tue 11/3/15**		**Phanideep**
29		6.2.1	Eclipse Plugins - 1	1 day	Tue 10/20/15	Tue 10/20/15		
30		6.2.2	Eclipse Plugins - 2	1 day	Tue 10/27/15	Tue 10/27/15		
31		6.2.3	Emacs	1 day	Tue 11/3/15	Tue 11/3/15		
32		**7**	**Assignments set 3**	**10 days**	**Tue 10/20/15**	**Mon 11/2/15**	**6**	**Phanideep**
33		7.1	Assignment 5	5 days	Tue 10/20/15	Mon 10/26/15		
34		7.2	Assignment 6	5 days	Tue 10/27/15	Mon 11/2/15		
35		8	MidTerm	0 days	Thu 10/22/15	Thu 10/22/15		Phanideep
36		**9**	**Testing**	**15 days**	**Fri 11/13/15**	**Thu 12/3/15**		
37		**9.1**	**Lecture**	**11 days**	**Thu 11/19/15**	**Thu 12/3/15**		**Zheng**
38		9.1.1	Unit testing	1 day	Thu 11/19/15	Thu 11/19/15		
39		9.1.2	Testing modules	1 day	Thu 11/26/15	Thu 11/26/15		
40		9.1.3	Types of testing	1 day	Thu 12/3/15	Thu 12/3/15		
41		**9.2**	**Lab**	**1 day**	**Tue 11/17/15**	**Tue 11/17/15**		**Phanideep**
42		9.2.1	Junit	1 day	Tue 11/17/15	Tue 11/17/15		
43		**10**	**Assignments set 4**	**10 days**	**Tue 11/17/15**	**Mon 11/30/15**	**8**	**Phanideep**
44		10.1	Assignment 7	5 days	Tue 11/17/15	Mon 11/23/15		
45		10.2	Assignment 8	5 days	Tue 11/24/15	Mon 11/30/15		
46		**11**	**Maintainence**	**10 days**	**Fri 12/4/15**	**Thu 12/17/15**		
47		**11.1**	**Lecture**	**4 days**	**Thu 12/10/15**	**Tue 12/15/15**		**Zheng**
48		11.1.1	Version control	1 day	Thu 12/10/15	Thu 12/10/15		
49		**11.2**	**Lab**	**1 day**	**Tue 12/15/15**	**Tue 12/15/15**		**Phanideep**
50		11.2.1	Subversion, GIT	1 day	Tue 12/15/15	Tue 12/15/15		
51		**12**	**Assignments set 5**	**5 days**	**Tue 12/1/15**	**Mon 12/7/15**	**10**	**Phanideep**
52		12.1	Assignment 9	5 days	Tue 12/1/15	Mon 12/7/15		
53		13	Final Exam	0 days	Thu 12/17/15	Thu 12/17/15		Phanideep