Answer 1(a):-According to Fred Brooks the 4 major challenges of software system are as follow:-

1. Complexity:-

The key factors which contribute to software complexity are as below:-

- Their elements/subparts are distinct.
- Due to large number of states.
- Communication among the various elements of software.

All the elements of software are distinct i.e. no two elements are same. For ex: In a simple calculator there are distinct functions for addition, Subtraction and other calculations. In opposite to this same function or structure is executed numerous times on the same RAM.

In running software there are n number of states which leads to making it difficult to understand and test. Where as in a mechanical machine like washing machine there are very few states like **Wash**, **DRY** and **SPIN** and this makes it easier for all the scenarios to be tested. In a software as there are numerous states, there may be a possibility that few test scenario are left not tested. This happens because those scenarios occur rarely.

Another key factor which adds on to complexity is the non-linear interaction amongst the various elements of software. In a simple washing machine its elements interacts in a sequential manner whereas in software elements interact in a non sequential manner due to its numerous states.

In today's fast growing demands of technology there is a need of complex software so that it can be compatible with different platforms and devices. Therefore the complexity of software is an essential property but nothing comes for free in this world. Complexity leads to many technical and management problems like difficulty in communication among teams who work on developing those complex software, product errors, over costing, delays, less understandability, less reliability, less security etc.

2. Conformity:

Software can't be developed in seclusion. Although many people believe that conformity can put limitation on the creativity but still software must be written following certain rules, standards, Hardware requirements and laws.

For ex: Computer Languages have their own predefined syntax and formats which a coder/programmer needs to follow to develop software. For instance executing the cobol program you need to run 2 separate JCLs, Compile JCL and Run JCL. Firstly for compiling an error free program and second for generating the desired output.

For developing a banking software, government policies like hiding private information (for ex: SSN number) of their customers must be considered.

This complexity cannot be reduced by scaling up or redesigning the software, all the attributes of this complexity must be included while designing the software.

3. Changeability:

Software often faces the problem of changes even if it is delivered in the production. But Software is inherently flexible where physical products like buildings, cars are not as it involves a high cost. The main factors to influence these changes could be:-

1) Challenges for the future are not foreseen in the design phase of the software.
2) Changing consumer needs
3) Ever growing Technical revolution.

As software has many states so there is a very high possibility that all test scenarios are not tested as they occur rarely. Thus in future if a rare test scenario occurs then changes in software will be required. A very famous example **is** Y2K bug**. P**rogrammers **did** not foresee the challenge of changing dates at the end of the century and later on programmers come up with various programming solutions like date expansion, date partition etc to solve this problem

The major driving force for change in software comes from the People who use it. They find new functionalities to be added on the software. For ex: whatsapp offered voice messaging and calling on its later versions.

Another factor which contributes to change in software is the changing environment. With the evolution of technology the platform/OS on which soft wares are written or the other devices like hard disk, display, printer changes. Therefore software should be changed/updated to make it compatible with the changing environment.

4. Invisibility:

Software is invisible in nature and can be visualize in term of logic and code which is very difficult to understand for a user. Entities like Building, automobile can be easily visualized by their Architect, one can draw a map or blueprint with the help of diagrams and directions. They have specific defined attributes like size, weight, and color etc. But in contrast in software it is difficult to visualize, there are many ways to visualize the same design .we can create its visibility with helps of algorithm, flowcharts and UML which are difficult to understand for a common user.

Answer 1(b):-

When I join the Software Industry 6 years back I was in the quest of a technology which really give me a platform to establish a approach in the solving the Software problems. My Learning and experience in the industry led me to the conclusion that Mainframes are Silver Bullets in my Software journey.

To illustrate my point below are few examples:

1. **In terms of Changeability:** Each generation make mainframe more secure, stable and compatible of all computing platform. Being Legacy Systems IBM Mainframe have always welcome changes. With Introduction of CICS (reliable Online System) in 1960s mainframe added new dimension to Batch and online systems. Mainframe are very adaptive to tools like CA7, CMAN, File manger along with database like DB2(SQL based),IDMS(Network System),VSAM. Now a day's Mainframe even support open source databases like Oracle.

   Tools like IBM debugger, OPC, CA7 added in the technology to make it more stable and secure and error free.

2. **In terms of Complexity:**

Since 1952 mainframe system are able to provide great Stability to Market's80% structured data which includes government, healthcare, Finances, Banking and various other fields. But at the same time cost of maintenance is decreased whereas the overall performance is increase by 70%.These legacy system maintain the whole life cycle of project and maintain security, convenience and accessibility to the various teams for their respective phases. Mainframes provide a common platform for software designed for high speed, optimal performance and integration with maximize business value.

3. In terms of Invisibility: Now a days with the introduction of evolution of internet most of the front end GUIs are developed by the languages like Java,.Net,Php etc. whereas Mainframe systems are invisible for the common user. But everyone is using Mainframe in his daily life. For example when you goes to the market and do the payment by Credit card (you are indirectly interact with Mainframe for transaction). In the software development teams can use techniques like Flowchart, UML , Algorithms to achieve the visibility for the flow of process

4. **In terms of Conformity:-**There is famous quote by John F Kennedy"Conformity is the jailer of freedom and the enemy of growth."

Critics often complain that Mainframes represent old ways and conformity. Mainframes have limited freedom and always complies with requirements of a specification, contract or regulation. Also with ever-growing technology Mainframes are start to support many open source tools without compromising on security of data.

Answer 2:-

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

Elements	Elements Name
Project Summary	Software Methods and Tools
Summaries	Meeting, Project Planning, Project Analysis & Design, Project Architecture, Project Implementation, Project Testing, Project Maintenance, Project Evaluation
Recurring Tasks	Lectures, Labs. These Tasks will frequently recurred on almost every week on Thursday and Tuesday **except Spring breaks**.
Tasks	Course Introduction, Software Development Process and Activities, Microsoft Project, UML Modeling I&II, Eclipse Plugin-ins I,II,SAD-I,SAD-II, Archstudio, IDE & Eclipse, Unit testing, JUnit, TBD,Version control, subversion, GIT –I&II
Milestones	Assignements 1-8,Midterm Project Evaluation, Final project Evaluation
Relationship	After completion of specific task/tasks you need to submit assignments. Ex:Currently I am doing this assignments after completion of Microsoft Project task.
Resources	Prof Y Zheng, Muying Cue, Varun Puri. All tasks are assigned to these resources. Assignments will be covered by Varun Puri

ID	WBS	Task Mode	Task Name	Duration	Start	Finish	Predece	Resource Names
0	**0**		**Software Method and Tools**	**74 days**	**Tue 1/19/16**	**Thu 5/5/16**		
1	**1**		**Meeting**	**73 days**	**Tue 1/19/16**	**Tue 5/3/16**		
2	**1.1**		Lecture	70 days	Tue 1/19/16	Thu 4/28/16		Prof Y Zheng
18	**1.2**		Lab	68 days	Tue 1/26/16	Tue 5/3/16		Muying Che
33	**2**		**Project Planning**	**9 days**	**Tue 1/19/16**	**Mon 2/1/16**		
34	2.1		Course Introduction	1 day	Tue 1/19/16	Tue 1/19/16		Prof Y Zheng
35	2.2		Software Development Process	1 day	Thu 1/21/16	Thu 1/21/16		Prof Y Zheng
36	2.3		Microsoft Project	1 day	Tue 1/26/16	Tue 1/26/16		Muying Che,Varun Puri
37	2.4		Assignment 1	0 days	Mon 2/1/16	Mon 2/1/16	36	Varun Puri
38	**3**		**Project Analysis & Desig**	**17 days**	**Thu 1/28/16**	**Mon 2/22/16**		
39	3.1		UML Modeling I	1 day	Thu 1/28/16	Thu 1/28/16		Prof Y Zheng
40	3.2		UML Modeling II	1 day	Thu 2/4/16	Thu 2/4/16		Prof Y Zheng
41	3.3		IDE & Eclipse	1 day	Thu 2/11/16	Thu 2/11/16		Prof Y Zheng
42	3.4		Eclipse Plug-ins I	1 day	Tue 2/16/16	Tue 2/16/16		Prof Y Zheng,Varun Puri
43	3.5		Eclipse Plug-ins II	1 day	Thu 2/18/16	Thu 2/18/16		Prof Y Zheng,Varun Puri
44	3.6		Assignement 2	0 days	Mon 2/8/16	Mon 2/8/16	39	Varun Puri
45	3.7		Assignement 3	0 days	Mon 2/22/16	Mon 2/22/16	40	Varun Puri
46	**4**		**Project Architecture**	**6 days**	**Thu 2/25/16**	**Fri 3/4/16**		
47	4.1		Software Architecture and Design I	1 day	Thu 2/25/16	Thu 2/25/16		Prof Y Zheng
48	4.2		Software Architecture and Design II	1 day	Tue 3/1/16	Tue 3/1/16		Prof Y Zheng
49	4.3		ArchStudio	1 day	Thu 3/3/16	Thu 3/3/16		Prof Y Zheng
50	4.4		Assignment 4	0 days	Fri 3/4/16	Fri 3/4/16	47,48	Varun Puri
51	**5**		**Project Implementation**	**21 days**	**Thu 2/11/16**	**Thu 3/10/16**		
52	5.1		IDE & Eclipse	1 day	Thu 2/11/16	Thu 2/11/16		Prof Y Zheng
53	5.2		Eclipse Plug-ins	3 days	Tue 2/16/16	Thu 2/18/16		Muying Che,Varun Puri
54	5.3		Assignment 5	0 days	Mon 3/7/16	Mon 3/7/16	41,53	Varun Puri
55	5.4		Assingment Demo	1 day	Thu 3/10/16	Thu 3/10/16		Prof Y Zheng,Varun Puri
56	**6**		**Project Testing**	**6 days**	**Tue 3/22/16**	**Mon 4/4/16**		
57	6.1		Unit Testing	1 day	Tue 3/22/16	Tue 3/22/16		Prof Y Zheng
58	6.2		JUnit	1 day	Thu 3/24/16	Thu 3/24/16		Varun Puri,Prof Y Zheng
59	6.3		Assignement 6	0 days	Mon 4/4/16	Mon 4/4/16	57,58	Varun Puri
60	**7**		**Project Managment**	**17 days**	**Thu 4/7/16**	**Mon 5/2/16**		
61	7.1		TBD	1 day	Thu 4/7/16	Thu 4/7/16		Prof Y Zheng
62	7.2		TBD	1 day	Tue 4/12/16	Tue 4/12/16		Prof Y Zheng
63	7.3		Version Control	1 day	Thu 4/14/16	Thu 4/14/16		Prof Y Zheng
64	7.4		Sub Version	1 day	Tue 4/19/16	Tue 4/19/16		Prof Y Zheng
65	7.5		Assignment 7	0 days	Wed 4/20/16	Wed 4/20/16	61,63,6	Varun Puri
66	7.6		GIT-I	1 day	Tue 4/26/16	Tue 4/26/16		Prof Y Zheng,Varun Puri
67	7.7		GIT-II	1 day	Thu 4/28/16	Thu 4/28/16		Prof Y Zheng,Varun Puri
68	7.8		Assignment 8	0 days	Mon 5/2/16	Mon 5/2/16	66,67	Varun Puri
69	**8**		**Poject Evaluation**	**32 days**	**Thu 3/17/16**	**Thu 5/5/16**		
70	8.1		Mid Project Evaluation	0 days	Thu 3/17/16	Thu 3/17/16		Varun Puri
71	8.2		Final Project Evaluation	0 days	Thu 5/5/16	Thu 5/5/16		Varun Puri