**Question 1: Explain why change is inevitable in complex systems and give examples (apart from prototyping and incremental delivery) of software process activities that help predict changes and make the software being developed more resilient to change.**

Change is inevitable in complex systems because of three reasons: business changes, new technologies and changing platforms. The business changes are often generated and requested by stakeholders, which is a group of people who have significant influence and power in the software project development or management. This causes the system requirements change: stakeholders can propose adding new requirements, deleting or changing existing requirements and guarantee they will be executed. The new technologies allow developers and stakeholders using new views to look at the systems and new methods to achieve the implementations. In both understanding and implementation levels, the improving implementations are parts of the system changes, which is an inevitable result of the development of technology. The new platforms and the requirement of changing platforms are other reasons that cause software project changes. With the development of software technologies, more and more platforms and development environments are being created and used in the software projects. It is inevitable that some modules and functions are required to be changed in order to fit the new platforms, whether for refactoring reasons or new requirements.

The software process measurement and analysis are two activities that can help predict changes and increase software resilience to changes. The process measurement is a process that can help decide if the resilience improvements are effective, by selecting one or more attributes from the system, measuring them and then creating a standard baseline for changes measurement. The process analysis is a process that identifies weaknesses and bottlenecks of the project process, which includes changes in the systems. It can also predict the changes by processing models and descriptions.

**Question 2: Apart from the challenges of heterogeneity, business and social change, trust and security, please identify other problems and challenges that software engineering is likely to face in the 21st century (think about the environment.**

New technologies change the implementation environments. With new programming languages and tools being developed, many existing implementations can be improved in better ways, as well as the unresolved problems can be solved in efficient ways. This requires software engineering to be flexible, adaptive and evolutionary in order to be ready for incoming changes, which requires many time and effort investments.

New platforms change the development and maintenance environments. Unlike old times, everyone uses mobile devices such as smartphones, tablets and laptops nowadays. With the increasing amount of mobile platforms and the needs of users, more and more applications have multi-platform requirements in the 21st century. It is a huge challenge for software engineering because some old application’s design may not fit the new platforms due to the different work process, while some application may take a lot of time and money to do the refactor and redeployment. Even for new applications, the cost of developing a multi-platform project is higher than a single platform project because there is much more work and things to be considered and implemented. Therefore, the new platforms are another challenge for software engineering.

**Question 3: Discuss the differences between software process metrics and software project metrics. When and why would each be used?**

The differences between software process metrics and software project metrics are the focuses. The software process metrics focus on analyzing the facts during the development process, such as process activities such as initializing, coding, testing, etc. The software project metrics focus on describing the attributes of the project, such as schedule, people, cost, etc. When we want to analyze one or more periods in a software development, the software process metrics work because they can be used to analyze different software phases and activities. However, if we want to study the software as a whole, the software project metrics are the best choice because they tell the attributes of the software at a general level. Unlike software process metrics, the software project metrics focus the software itself, instead of going through and analyzing different phases and activities.