# Software Engineering Workshop 1 – Exercise 4

## Question 1:

Jan Bosch's presentation about the innovation cycle and data-driven features identification for product improvement and evolution highlights 3 key points. The first key point is that innovating new systems, solutions and services at an unprecedented rate is critical for the survival of systems. This is because the technological world is moving from product to the services the product provides. Therefore, to ensure that customers are engaging and interested in your product new services and functionalities should be continuedly added and updated. The next key point is that embracing data-driven, instead of opinion-driven, decision making accelerations innovation. By doing this the overall cost of the project is reduced. This is because companies can easily identify actives that are not useful to the user/system and focus on developing functionalities used by customers. Furthermore, companies need to know the customer needs before the customer knows they need it. This is achieved by continuously testing products and services with the customer to determine which they respond well to and which they do not. Bosch emphasises that successful companies are those who can run the largest number of experiments. The final key point is that systems engineering should shift from "built to last" to "built to evolve". This is about being able to add features and services to systems instead of building a system that only produces one feature but will last forever.

## Question 2:

Post-deployment data collection and usage form the basis of ensuring the continuous improvement of existing products and systems and aids in future innovation and product development. As technology and trends are constantly changing, current systems must be able to adjust, adapt and update according to customer behaviours, needs, feedback, and emerging requirements. As these systems are connected to the internet, data can begin to be collected immediately when the system is used. Typically, "bug" type data is collected, for instance, information on system restarts, system outage, faults, and other failures, as well as system run-times, update times and success rates. This collected data provided insight into the system and helps to quickly identify any issues with the system, as well as, helping to validate, prioritize, optimize, and deliver functionalities of the system. Currently, this data is not being optimised and is only being used for troubleshooting purposes. Furthermore, companies struggle to integrate, communicate and visualize the collected data in a way to make it accessible for employees. Therefore, the data is primarily used as input for the next pre-development phase and troubleshooting.

The post-deployment usage framework outline how companies can optimise their post-deployment product data. The first level is operational data which represents data collected about the system performance. This provides companies with a high level of insight to how the system is operating in real-time. Next is diagnostic data which represents data utilised in troubleshooting and system maintenance activities. At this level, companies should ensure that effective documentation and tracing of troubleshooting and maintenance processes occur. The next level is feature usage which represents data that aids companies in understanding the usage of individual features. This is achieved through a high-level of data analysis which customer usage patterns of specific features can be distinguished. The next two levels, feature improvement and new feature development, represents data collected that supports the continuous improvement of current functionality and the development of new features. These levels are reached through continuous deployment, A/B testing practices, rollback mechanisms, and feature experimentation.

**Question 3:**
The elicitation techniques we have chosen to undertake for our systems requirements are questionnaires and systems interface analysis. Both of these elicitation techniques only partially focus on data-driven features identification but neither focuses on post-deployment data collection. Data-driven features identification is used in system interface analysis because as data received from the interfacing system is used to develop requirements. Questionnaires can be used to collect data from customers about their wants and needs for the student life system, however, this data only accounts for current requirements of the customer and does not identify any future user needs. Furthermore, the data collected from questionnaires is subjective meaning that the data is bias to the responses of respondents and may accurately represent the whole intended audience. In terms of post-deployment data collection both of these elicitation techniques focus on gathering requirements before the creation of the system not during the release. However, questionnaires can be conducted after the release of the system as a mean for post-deployment data collection.

Data-driven approaches to identifying requirements/features and post-deployment data collection can be used for continuous improvement and evolution of the student life system. By collecting data post-deployment, developers will be able to troubleshoot the system based on data collected from system restarts, system outage, faults, and other failures. Furthermore, by collecting post-deployment data developers will be able to determine which features and pathways, within the system, are being utilised the most by customers. Therefore, developers can spend more time developing and upgrading these highly used features rather than wasting recourses and time on developing features that are rarely used. Likewise, we may be able to elicit future requirements for customers by completing customer experimentation. In these experiments we should present a focus group with a range of possible new features for the student life application, this will help to determine which features customers would utilise, and which should not be incorporated into the system.