**Homework 2: Software Processes**

**Problem 1**

*Suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems. Explain your answer according to the type of system being developed:*

* *A system to control anti-lock braking in a car.*
* *A virtual reality system to support software maintenance.*
* *A university accounting system that replaces an existing system.*
* *An interactive travel planning system that helps users plan journeys with the lowest environmental impact.*

1. Car anti-lock braking system.

* **Waterfall model:** We should expect the highest reliability for a control system within a vehicle because there is a human life risk involved. The predictable nature of the waterfall model guarantees testing and maintenance before adding a new feature, thus maximizing safety.

1. Virtual reality support system.

* **Incremental development:** Virtual reality space is considered a bleeding-edge technology with rapid evolvement in the industry. Incremental development is suitable as it can deliver new functionality immediately, and existing features can change dramatically.

1. University accounting system.

* **Integration and configuration:** An accounting system typically communicates with third-party entities such as banking services and employee payroll management. It could also be a part of a more substantial system like Enterprise Resource Planning (ERP). Being connected with many services requires high configuration, which this methodology excels.

1. Interactive travel planning system.

* **Hybrid (incremental development, integration & configuration):** Incremental development is the primary method because an interactive interface visualizing the user's carbon footprint sounds like a new concept that requires constant customer feedback. Moreover, this system should adapt to the most recent environmental data affected by climate change. As a secondary methodology, utilization of integration and configuration would help the system to communicate with many airlines’ Application Programming Interface (API).

**Problem 2**

*Incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. Discuss.*

Incremental development is flexible as it promotes quick modification in no specific order. The lack of structure is suitable for a customer whose idea is not yet clearly outlined because the engineering team can start working on the current instructions and worry about the revision later. This methodology can deliver a working concept sooner, albeit with unfinished parts.

**Problem 3**

*Explain why software testing should always be an incremental, staged activity. Are programmers the best people to test the programs that they have developed?*

A programmer may perform a standard operation check when testing software. However, with a limited dataset, this type of test lacks depth and disregards edge cases. In contrast, incremental testing provides ordered tests that can track a more targeted issue, such as integration, unit, and performance. In my opinion, a programmer is a competent tester since the author is familiar with his/her writings. However, independent testers are theoretically the best people to perform the checks as they bring new perspectives and are not conflicted by personal relationships presumably rife within the engineering team.

**Problem 4**

*Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.*

User requirements refer to desired functionalities and goals expressed by the customer. System requirements, on the other hand, translate these needs into technical specifications for the system to be built upon(JD, 2008). Recognizing this distinction is crucial, as each document serves a different purpose for different stakeholders. Separating them allows us to create a clear roadmap for software development, thereby minimizing misunderstandings and other unintended consequences.

**Problem 5**

*Suggest five possible problems that could arise if a company does not develop effective configuration management policies and processes.*

* 1. **Manual configuration:** Configuration management automates the maintenance process of a system. Manual configuration of every component would be necessary for each requirement change without automation (Upguard Team, 2022).
  2. **Delayed communications:** Configuration management tools provide real-time change detection. Untimely communication between all parties could result in stalled or failed implementations.
  3. **Wasted resources:** An internal software team may find a new version of a library to be flawed during an update, thus wasting time and resources for an undesired result. Most configuration managers solve this issue with the ability to revert to the previous state at any development point.
  4. **Unintended consequences:**Determining the specific components affected by adjusting a configuration in a system can be challenging. Configuration management alleviates this pressure by tracking changes every time a configuration is modified.
  5. **Inefficient resources:** Large software projects tend to repeat work unnecessarily. For example, they may have duplicate assets or redundant implementation. It takes away precious resources and should be managed by a configuration management tool instead.

# Bibliography

JD. (2008, June 9). *Why Differentiate User Requirements vs. System Requirements?* Retrieved from Shaping Software: https://shapingsoftware.com/user-requirements-vs-system-requirements/

Upguard Team. (2022, August 01). *What Is Configuration Management and Why Is It Important?* Retrieved from UpGuard: https://www.upguard.com/blog/5-configuration-management-boss/