CS 487 – SOFTWARE ENGINEERING

Week 7 Engagement – Artificially Intelligent Systems

Suhas Palani **A20548277** spalani3@hawk.iit.edu

Risk Management in Software Development

In software engineering, risk management is critical when deciding how to allocate resources. To direct the engineering process, risk exposure is commonly characterized as a mix of **likelihood** and **impact**. Risk can be minimized by lowering the possibility that an issue will arise or by limiting its possible consequences.

For instance, **redundant engineering** may be used to create backup systems to ensure reliability. However, it is essential to weigh the expenses of this redundancy against the benefits, as having too many backup systems may have diminishing returns. Ensuring that the cost of risk reduction is justified by its benefits is the goal.

Iterative Development and Prototyping

A preferred method for contemporary software engineering is the iterative development process. It prioritizes regular user input and rapid prototyping over the waterfall model's inflexible phases. The system may adapt depending on real-time feedback by iteratively generating prototypes and testing them with users. This lowers the chance of providing a product that doesn't satisfy the demands of the consumers.

Using iterative methodologies, issues may be identified early in the development process and fixed continuously, resulting in higher system quality and user satisfaction. Because frequent user engagement increases the likelihood that the product will meet user expectations, this strategy also improves user acceptability testing.

Artificial Intelligence and Automation

Artificial intelligence (AI) is increasingly essential for automating processes that once required human input as software systems advance. AI systems are supposed to simulate degrees of **awareness**, allowing them to detect their surroundings, analyze data, and make decisions autonomously.

In many applications, automating awareness and decision-making improves consistency and efficiency, but it also poses new difficulties. Intelligent artificial intelligence (AI) must be able to identify **exceptions**—situations that depart from the norm—and react appropriately to sophisticated systems, such as those used in air traffic control or healthcare. To avoid failure or mistreatment, systems must be designed to manage these exceptions properly.

Security and Safety in Software Engineering

To create dependable and trustworthy systems, **nonfunctional requirements** like security and safety are essential. Data integrity and system protection are ensured by security methods including **multi-factor authenticati**on and **layered security**. Effectiveness and cost must be balanced in the design of these measures since too complicated security systems might reduce usability and increase development expenses.

Safety is just as important, especially for mission-critical systems where a malfunction might have severe repercussions. These systems need to be built with operational safety in mind, reducing the possibility of security lapses or malfunctions. Since the goal of safety and security systems is to minimize the possibility and consequence of a failure, the idea of **risk exposure** is pertinent in this context.

Awareness and Exception Handling in Automation

The term "awareness" in software systems describes the system's capacity to detect its surroundings and identify unusual circumstances. This is a crucial component in creating autonomously operating automated solutions. Exception handling becomes an essential component of this process as systems must be able to recognize problems and take the necessary corrective action when they arise.

Systems rely on preprogrammed reactions and **memory banks** with information about possible circumstances to handle this. Systems can function more efficiently even in unexpected contexts if they can identify exceptions and have predetermined responses.

The Role of Automated Systems and Decision-Making

With developments in technology, automated systems are now capable of more complex **decision-making**. Without the need for human involvement, these systems are capable of evaluating complicated data sets, including risk considerations, user interactions, and environmental variables, and making defensible judgments. While automating these decision-making procedures increases productivity and lowers human error, it also necessitates rigorous system building to guarantee that it can manage unforeseen circumstances.

Automated systems are being used more and more to carry out essential activities for a variety of industries, including healthcare and banking, as their capabilities continue to expand. For these systems to be implemented successfully, they must be strong, secure, and dependable