[**Homework 19**](https://github.com/hendraanggrian/IIT-ITM511/blob/assets/assignments/hw19.pdf)**: Systems engineering**

**Problem 1**

*Give two examples of government functions that are supported by complex sociotechnical systems and explain why, in the foreseeable future, these functions cannot be completely automated.*

1. **Internal Revenue Service (IRS) — Tax & revenue collection:** The IRS manages a person's tax based on tax filing and regional law. Government employees audit each tax file to prevent misreporting or legal misrepresentation. An automated system could miss a few details because it is not suited to adapt to personal circumstances.
2. **Federal Emergency Management Agency (FEMA) — National disaster response:** FEMA and possibly national guards are the first responders when tragedies occur. They coordinate with local agencies to aid with infrastructure damage and refugee needs. Delegating this task to an automated system could risk an inadequate recovery strategy in disastrous events.

**Problem 2**

*Explain briefly why the involvement of a range of professional disciplines is essential in systems engineering.*

System engineering study concerns building the entire software system instead of an individual component. It involves software engineering and dealing with external aspects beyond engineering like safety and usability. Because system engineering has many responsibilities, professional disciplines bring domain expertise essential to the system's operation.

**Problem 3**

*Complex sociotechnical systems lead to three important characteristics. What are they? Explain each in brief.*

1. **Emergent properties:** In system engineering, emergent properties like security are characteristics of interactions among system components (Sommerville, 2016). They can be assessed after establishing their relationship with each other.
2. **Nondeterministic result:** The program may not produce the same output given identical input because human operators interfere with the process. New relationships created during deployment also affect its emergent properties.
3. **Subjective criteria:** The criteria of success depend on the relationship among components as much as the system itself. The success criteria should adapt to the changes in the organizational leadership.

**Problem 4**

*Why is system integration a particularly critical part of the systems development process? Suggest three sociotechnical issues that may cause difficulties in the system integration process.*

A software system is built from smaller components with distinctive objectives and technologies. Integrating them into a single unit is challenging because of potentially inconsistent data formats or communication protocols. To create a unified user experience, system integration has to balance the workload among software components and resolve sociotechnical problems.

Several sociotechnical issues that may arise in the development process:

1. **Software interoperability:** Incompatibility in the selected components because of conflicting standards or interfaces. In more alarming scenarios, a major rewrite is necessary for correct integration.
2. **Organizational culture:** The working environment can impact the final product. For example, a company's tendency to resist change is making them susceptible to keep using legacy technology.
3. **Communication gaps:** Large companies regularly employ international workers to stay competitive. Miscommunication is a frequent concern in a software team with diverse language and cultural backgrounds.

**Problem 5**

*Why is system evolution inherently costly?*

The evolution of a software system is a process of modifying or migrating software components based on changing user requirements. This process is substantial and expensive since an architectural change can bring cascading effects to other dependencies. Moreover, high-longevity systems often rely on legacy code that no longer fits modern guidelines. To evolve a software system, the organization requires an extra investment for additional employee allocation and expertise training.

# **References**

Sommerville, I. (2016). Software Engineering. In *Software engineering diversity* (10 ed., pp. 558–559). Pearson Education.