Unit 3

Introduction to Information Systems

Discussion Forum – Example of Information System Failure Summary

Over three weeks, consideration was given to defining information systems (IS). A high-level definition of an information system is a system that processes data with context to provide information. Given this definition, information systems are pervasive in our modern society wherever electronic data processing is required.

Looking into reported IS failures, the single reason information systems fail is an unknown operating condition, whether the condition originates from data, device or human involvement. Many of the discussion points referred to a lack of testing, monitoring or extremely poor or non-existent auditing processes. Therefore, a takeaway is that data is a fundamental component; designers must ensure that all conditions supporting said data (storage, transport, security, transformation, traceability) are defined and well-documented. Moreover, such processes must be alive and well established long before deploying any IS for public use. According to Gauld (2006), IS failures are common, with most failures (84 per cent) occurring in the public sector. Therefore, all information systems stakeholders must consider the value of the underlying data and associated processes. Steward (2021) considers that IS security is often an afterthought; so, incorporating security must occur early on within each IS design stage when utilising a Software Development Lifecycle (SDLC) methodology.

During the Design Phase of the SDLC methodology, IS designers often leverage Object-Oriented designs to model real-world “objects” as it is easier to model based on real-world concepts (object) and interactions (behaviours) than it is to build systems using abstract notions. However, a quote from C.A.R Hoare (as cited in Pressman:218, 2010) nicely sums up the difficulty of designing systems: “There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.”

Lastly, I consider that IS failures reflect the difficulty for humans to test every possible permutation on every possible hardware configuration. As a result, IT professionals carry greater responsibility in reducing possible failures. Thankfully, Artificial Intelligence and Machine Learning are two emerging trends that may come to our aid (Soto et al., 2019)

References

Gauld, R. (2006) Public sector information system project failures: Lessons from a New Zealand hospital organisation. Government Information Quarterly 24(1): 102-114. https://doi.org/10.1016/j.giq.2006.02.010

Pressman, R.S. (2010) Software Engineering: A Practitioner's Approach. 7th Edition. New York: McGraw-Hill.

Soto, J.A.C, Tavakolizadeh, F & Gyulai, D (2019) An online machine learning framework for early detection of product failures in an Industry 4.0 context. International Journal of Computer Integrated Manufacturing 32(4-5): 452-465. https://doi.org/10.1080/0951192X.2019.1571238

Steward Jr., C, Wahsheh, L.A, Ahmad, A., Graham, J.M., Hinds, C.V., Williams, A.T. & DeLoatch, S.J. (2012). "Software Security: The Dangerous Afterthought": 2012 Ninth International Conference on Information Technology - New Generations. Las Vegas, NV, USA, 16-18 April 2012. DOI: 10.1109/ITNG.2012.60.