

It425-1403a-01 : systems analysis, design and integration

[Technology, Information Technology](#)



Running Head: Systems Analysis, Design and Integration. College System Development Life Cycle (SDLC) is the development and implantation of an information system in a multi-step process involving phases from initiation through analysis, designing & implementation to maintenance and disposal phases. SDLC models constitute series of phases and integrate security of information to guarantee high integrity any information to be transmitted, processed or stored on the system. By including management of risks during the development of systems, institutions can balance the need for protecting important assets and information with costs associated to security control and mitigation plans as the system is developed (Goldman, 2012).

The very first phase in SDLC models is initiation. Initiation involves the expression of a need for the system and documentation of the specific purpose as well as high level specifications for the system. Development or acquisition is the next phase- it comprises designing, purchase, programming and development of the system. Sub-phases here include system development and acquisition cycles. Implementation or assessment is the 3rd phase involving installation and it follows initial tests of the systems performance. The fourth phase involves the system performing the intended purpose and thus is referred to as operations or maintenance. The sunset phase involves the disposal of the system on completion of its transition to a new computer system (Radack, 2009).

SDLC models are ultimately cost effective as it is possible to identify and mitigate security issues early. Risk management is applied in a timely fashion thus facilitating informed executive decisions. Crucial decisions concerning security of the system are made throughout development and

are used to inform management's considerations on security. Through SDLC models, improved integration and interoperability of the system is attained which would have been difficult if security was considered separately.

Rapid Application Development on the other hand is the process in which the entire cycle of developing a system is expedited. It therefore makes possible to fast develop quality system hence saves on resource use. RAD makes use of techniques and tools which are automated hence restructuring the entire process of constructing information systems. It effectively replaces coding and hand design which largely depend on skills by different persons with automated processes of design that are extremely stable. RAD is fast and less error prone, therefore more capable. As compared to traditional cycles such as SDLC, RAD is less rigid thus meets the dynamic needs of modern business. Crucial ingredients in RAD are the methodology, the people, management & tools. It combines the best techniques, uses prototypes that are evolutionary, makes use of workshops, implements time bound development processes and offers guide to success while outlining pitfalls to be avoided. The phases include planning of requirements, user design, construction and implementation of the system (Rouse, 2007).

Joint Application Development (JAD) is an interactive concept of system design development involving workshops to discuss and compile system information. It brings together developers, system users and other opinion in an environment of creativity. Phases include definition of the project, brainstorming on requirements by the user, preparing for JAD sessions, caring out as well as facilitating the sessions and coming up with a comprehensive document incorporating all decisions reached. It is less time

intensive, improves the quality of the final system, reduces costs, enhances education for users hence cultivating ownership and also reduces functional creep. Constant communication between developers, final users and the business ensures the system developed is easily understood, accepted and implemented (Yatco, 199).

Basing on my research on system development models I settle on a hybrid of models for my class work. This because of the many advantages and weaknesses inherent in the different specific models and as such combining them is more appealing to me. With a combination of models I will be able to take advantage of the strengths of the models and minimize the weaknesses hence come up with a more stable, versatile, secure, user-friendly and acceptable project.

References.

1. Blanchard, B. S., & Fabrycky, W. J.(2006). Systems Engineering and Analysis (4th ed.) New Jersey: Prentice Hall. p. 31
2. Eric H. Goldman. (2012). The Effect of Acquisition Decision Making on Security Posture. Information Management & Computer Security, Vol. 20 (5), pp. 350 – 363
3. Marakas, James A. O'Brien, George M. (2010). Management Information Systems (10th ed. ed.). New York: McGraw-Hill/Irwin. pp. 485–489. ISBN 0073376817.
4. Margaret Rouse. (2007). Rapid Application Development (RAD).
5. Mei C. Yatco. (1999). Joint Application Design/Application. University of Missouri-St Louis. Retrieved at <http://www.umsl.edu/~sauterv/analysis/JAD.html>

6. Peter Crock. (2008). Best Practices for Software Projects-JAD Sessions. Evaltech Research Group. Retrieved at http://www.evaltech.com/pm/JAD_session.pdf
7. Sherley Radack. (2009). The System Development Life Cycle (SDLC). National Institute of Standards and Technology. Available at http://csrc.nist.gov/publications/nistbul/april2009_system-development-life-cycle.pdf
8. Tommy Olsson, and Paul O'Brien. (2008). The Ultimate CSS Reference. Site Point Pty Ltd. Available at <ftp://91.193.236.10/pub/docs/linux-support/programming/HTML-CSS/%5BSitePoint%5D%20-%20Ultimate%20CSS%20Reference%20-%20%5BOlsson,%20O%60Brien%5D.pdf>