



System Development Life Cycle Models

Surya Lakshmisri

Wilmington University

SDLC seven phase model

The entire system development cycle is the base for all systems development and methodologies. There are several activities related to each phase in SDLC. SDLC seven phase model is a model that consists of seven stages. It consists of a systematic progression and a procedural checklist that are necessary for developing an IT system from commencement. The seven stages of this model include conception planning, planning and requirements definitions, design, development and testing, implementation, operations and maintenance and disposition (Cannon, 2009).

Conceptual planning is the first stage of this model similar to any other system's life cycle. During this stage, the system developer identifies the need to acquire and enhance a system, assesses the feasibility and the cost of the system and defines the risks and other various project planning approaches. The stage also involves the designation of the roles of the asset manager, system development agent, sponsor's representative and system support agent. In the second stage which is the planning and requirements definitions, the functional, support and training necessities are collected, defined and validated. The stage also involves the development of the initial life cycle management plans (Dora & Dubey, 2013). In the design stage which is the third stage of the system lifecycle, there is a translation of the functional, support and training requirements into preliminary and comprehensive designs. A preliminary design that acts as the high-level guide is also produced. The stage also involves making decisions on how the system would meet the functional requirements (Broad, 2013).

The fourth stage is the development and testing stage. This stage involves the developing or the acquisition of the system according to the detailed design specifications. After the development of the system, it is validated to ensure that it functions as expected and it satisfies the requirements of the sponsor. The phase also involves the acquiring, testing and integration of all the components of the system such as communications, procedures, and applications and associated documentation (Gary, 2009). After developing and testing the system, the cycle enters the fifth stage that is an implementation. This stage involves the installation of the new system in environment, training of users and data conversion. After the installation of the system, it handed over to the sponsor who evaluates the business processes. The problems identified during the implementation process are also solved during this stage. After the installation of the system and the training of the users, the system is fully operational. This is the sixth stage of the cycle which also involves maintenance of the system. During this stage, the system developer ensures that the system continues to meet the sponsor's needs by performing according to the sponsor's specifications. There are routine software and hardware maintenance and frequent upgrades to ensure effectiveness of the system in its operations. Similar to stage five, user training continues at this stage to introduce new features of the system to the current users or to accustom new users to the system (Cannon, 2009). IT software testing alludes to breaking down and grilling a product program to uncover mistakes and mystery issues. Software testing is anticipated to guarantee that an incorporated programming program performs to meet fulfillment limits, access, and quality protection. For an Information Technology framework to be finished, the product testing domain ought to not miss in the Software Development Life Cycle (SDLC). Under the necessities of SDLC, programming advancement is not supreme under it is exposed to the testing procedure. Characteristically, testing isn't performed to show a mistake free framework yet to build up a certainty divider that underpins the establishment and execution of the whole framework (Mohammad, 2015).

The last stage of this system model is the disposition. This is the end of the system's life cycle which involves the termination of the system cycle by preserving vital information for future access or reactivation. The disposition stage implies that the system has become outdated or is declared a surplus hence scheduled for shutdown (Broad, 2013).

SDLC four-phase model

This type of a system model has only four phases that include deciding, design, develop and demonstrate (Shikha, 2012). The decide phase involves three distinctive stages these include, business case, user requirements, and system specifications. The business stage involves identifying the reason of the user to require the system and what he or she wants to achieve using the system. During this stage, the system developer evaluates the current situation of the system, the business opportunity that the new system intends to solve, the various strategies that can be applied to develop the system and their feasibility and the preferred solution strategy. The developer also evaluates the costs and the benefits of the system together with the risks, assumptions and the possible constraints (Gary, 2009).

The next stage of the first phase is the user requirements. This stage involves the definition of the requirement needed to develop the preferred system in order to achieve the business opportunity or the sponsor's objectives. The stage involves the definition of the functional requirement, non-functional requirements and the design objectives. The functional requirements of the system include the purpose of the system e.g. to store customer records while the non-functional requirements include performance aspects of the system and development aspects of the system such as the time required to develop the system. On the other hand, design objective involves defining the features that would apply to the system (Aken, 2008).

The last stage of the first phase is system specification. During this stage, the focus is on the system being developed. The developer identifies the technical process needed to develop each system process, external interfaces or the requirements of the system to communicate with the outside such as screens and connections to other software systems (Massey & Satao, 2012).

The second phase of this model is the design phase. During this phase, the various requirements of the system are mapped into a comprehensive design. The phase involves coming up with the architecture of the system by defining the various component of the system and their interactions. The phase also involves designing how each of the components of the systems will work and communicate. The result of this phase is a comprehensive design (Shikha, 2012).

The third stage is the development phase. This stage involves the developing or the acquisition of the system according to the detailed design specifications. The components of the system are built according to the

specification of the sponsor (Wallin, 2002). After the development of the system, it is validated to ensure that it functions as expected and it satisfies the requirements of the sponsor. After the development of the system, the development cycle comes to its last stage that is the demonstrate phase. In this phase, tests are carried out on the newly developed system to ensure that it meets the needs of the sponsor by conforming to each of the designs, requirements and specifications as planned in the decide and plan phases (Gary, 2009).

Comparison between the SDLC seven phase model and the SDLC four-phase model:

Similarities

Both the SDLC seven phase and the SDLC four phase models are models that are applied in developing a system (Wallin, 2002). In both cases, the end product of the cycle is a newly developed system. The initiation of both the models involves identifying the need of having a system and evaluating the costs, feasibility and the risks associated with the selected system. In addition, the design phase of the SDLC four-phase model is similar to the development or the construct component of the SDLC seven phase model. In both models, the newly developed system is tested to ensure that it satisfies the sponsor's needs by being able to meet the objectives of developing the system (Shikha, 2012). Through SDLC methodology, Innovations have expanded themselves and along these lines new procedures have been grown, namely continuous integration (CI), continuous delivery (CDE) and continuous deployment (CD), helped by the entirety of the previously mentioned open source and restrictive arrangement. Computerization using SDLC is the essential necessity for DevOps, and DevOps' principle idea is "Automate everything". DevOps mechanization initiates from the coding on the designer machine until the code is run and afterward the application and framework are controlled underway (Mohammad, 2016).

Differences

The SDLC four-phase model shows a number of contrasting characteristics as compared to the SDLC seven phase model. Unlike the SDLC seven phase model that has seven distinct phases, the SDLC four-phase model has only four phases. This model combines phases one and two of the SDLC seven model to come up with the decide phase. Similarly, the model lacks the last three phases of the SDLC seven model that include implementation, operations and maintenance and disposition. Unlike in the SDLC seven model, the SDLC four-phase model does not involve training of the system users to acquaint them with the system (Shikha, 2012).

References

- Aken, A. (2008). CHUNK: An agile approach to the software development life cycle. *Journal of internet commerce*, 7(3), 313-338. Doi: 10.1080/15332860802250385
- Broad (2013) Risk management framework: A lab-based approach to securing information systems (pg 43) Newnes publishers, USA
- Cannon (2009) CISA Certified information systems auditor study guide (pg 267); John Wiley & Sons, USA
- Sikender Mohsienuddin Mohammad, "AUTOMATION TESTING IN INFORMATION TECHNOLOGY", *International Journal of Creative Research Thoughts (IJCRT)*, ISSN:2320-2882, Volume.3, Issue 3, pp.118-125, August 2015, Available at :<http://www.ijcrt.org/papers/IJCRT1133463.pdf>
- Dora, K. & Dubey, P. (2013). Software development life cycle (SDLC) analytical comparison and survey on traditional and agile methodology. *National monthly refereed journal of research in science & technology*.
- Gary (2009) Systems analysis and design; Cengage learning, USA
- Massey, V. & Satao, K. (2012). Comparing various sdlc models and the new proposed model on the basis of available methodology. *International journal of advanced research in computer science and software engineering*, Vol. 2 (4); 2277
- Shikha, (2012). A comparative analysis of different types of models in software development life cycle; *International journal of advanced research in computer science and software engineering*, Volume 2, Issue 5
- Sikender Mohsienuddin Mohammad, "**CONTINUOUS INTEGRATION AND AUTOMATION**", *International Journal of Creative Research Thoughts (IJCRT)*, ISSN:2320-2882, Volume.4, Issue 3, pp.938-945, July 2016, Available at :<http://www.ijcrt.org/papers/IJCRT1133440.pdf>
- Wallin (2002) Software development lifecycle models the basic types; Retrieved from; http://www.idt.mdh.se/kurser/ISD_cdt417/files/articles/paper5.pdf
- Sikender Mohsienuddin Mohammad, "DEVOPS AUTOMATION AND AGILE METHODOLOGY ", *International Journal of Creative Research Thoughts (IJCRT)*, ISSN:2320-2882, Volume.5, Issue 3, pp.946-949, August-2017 or Available at <http://doi.one/10.1729/Journal.24060>