

System Development Life Cycle

Many business company uses software systems or specific programs to help organize their organizations. System development life cycle is one of the common information systems in today's business industries. System development life cycle is a structured step-by step approach for developing information systems. SDLC involves 7 steps also calls phases in order to activate the information systems. The activity of SDLC is to determine the budget, gather business requirements, designing models, and writing detailed user documentation. The seven phases includes planning, analysis, design, development, testing, implementation and maintenance. In order for the information system to follow through, it's mandatory to use all seven phases. However, SDLC have many different versions of information systems, follow-up with different steps as well. We'll discuss what are the seven phase of SDLC and what the phases specializes in. Who uses the SDLC information systems and why. Along with that we will also discuss the different version of SDLC, difference between other versions and the pros and cons. Lastly, we will include companies and business testimonies of the different versions of SDLC along with the advantages and disadvantages of using the systems.

In order for the systems to be effective, it needs to engage the seven phases. The first phase will be "Planning"; this phase is to define the system to be developed. You must select a system for development or to determine which system is required to support the goals of the organizations. For example, our job is to help a mom and pops restaurant to become more of a modern restaurant. The mom and pop's restaurant does everything

by hand; from writing down the orders, inventory, and payroll. Our goal is to development a system that makes their organization more organize and easier for them. We'll build a system that will help them take order faster and processing it the kitchen rather running back to the kitchen and hand it to the chef. Along with that the system can maintain inventories. The system will be similar to other restaurants, for instance, Cheesecake Factory. Cheesecake Factory restaurant is an updated modern restaurant who uses SDLC system to organize the franchised organizations, such as touch screen computerize. The project plan also involves a time required to complete each activity.

The second phase is to analysis. Analysis phase involves end users and IT specialists working together to gather, understand, and document the business requirements for the proposed system.

The third and fourth phase is to design and development. Analysis phase- System development life cycle involves end users and IT specialists working together to gather, understand, and document the business requirements for the proposed system. Development phase is taking all of yours detail design documents from the design phase and transform them into an actual system. For instance, building the hardware and software for that specific system that is being assigned.

The fifth phase is testing. This phase is to verified that the system works and meets all the business requirements defined in all analysis phase. Performing the testing of the system it needs to be tested in different ways. Different testing methodologies are available to undo the bugs that were committed during the previous phases. Different testing tools and methodologies are already available. Some companies build their own testing tools that are tailor made for their own development operations.

The sixth and seventh phases are implementation and maintenance.

Implementation is you distribute the system to all the knowledge workers and they begin using the system to perform their everyday job. Provide training for the workers who will be using the new systems. It can be online training or workshop training. Using the mom and pops shop, this applies to them. Since, they employees there are not familiar with the new system. It's mandatory for them to get the proper training on how to access and use the systems.

Maintenance is the final phase of any system development. The maintenance phase is you or an IT specialist to monitor and support the new system to ensure it continues to meet the business goals. Building a help desk to support the system user or provide an environment to support changes such as any changes arises in the business environment, they must assess the input changes in the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the implementation period.

Many company uses SDLC to improve and organize their business. Highlighting a couple of companies who uses SDLC version or specializes in the field. Stylus Inc. is a company who installed information system to business. Their goal in the business is to take you request project and satisfied the objectives. Stylus Inc. develops custom software solutions for ERP, CRM and many more. They provide software development services.

There are numerous amounts of SDLC's, each type performing a different task for a specific job. With the many variations of this process each of which contains particular aspects that differ itself from the others. Choosing the correct SDLC to use for your

specific job is crucial. Given the vast amount of difference between these forms of SDLC and understanding the reasons for choosing a particular one to use is vital. The one of the different version is waterfall.

The Waterfall model is one of the most common and heavily used of the SDLC models. It is often referred to as the classic approach because it describes the development method which is linear and sequential. This method has the goals already set for each phase of the process. “Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back”¹. This quote describes the meaning behind the naming of this method being compared to a waterfall. As you can see once you have completed a phase there is no turning back, and this is the main con of this model. If there is a problem later found with the program you can't go back because of the regulations of the waterfall model's requirements. The main advantage of the waterfall model is that it allows managerial and departmentalization control. The forecasted schedule of the model has specific deadlines for each stage which must be meeting in the time period aloud for the step being performed. The process of development moves from the concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance, with each of the stages cohering to a stringent order of process which abides by the rule of no overlapping or iterative steps. One of the other main cons of the waterfall model is that it doesn't allow for much revision or reflection after the stages have been completed. With these problems many companies who are processing there

¹ Techtarget

program from a trial and error process should avoid using the waterfall method. Because of the inability to change there is little room for error for companies who are trying to develop a system which they are not exactly sure what the goals and objectives of the systems are.

The next popular version is spiral model. This SDLC system is a combination of the waterfall model and the prototyping model. This model is used for larger more expensive and complicated jobs for the development lifecycle. The advantages of using this particular model are that it is good at following budgets and schedules which are important issues for companies who need to follow these guidelines for business purposes. It also can manage the constant changes of the cycle which occur in the process of development, and the engineers who are using this process can get to work faster and more efficiently because of the faster process of development. The steps of this process differ from the normal SDLC. The spiral model system requirements usually involve interviewing a number of users representing all the external or internal users and other aspects of the existing system.

1. A preliminary design is created for the new system.
2. A first prototype of the new system is constructed from the preliminary design.
This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
3. A second prototype is evolved by a fourfold procedure: (1) evaluating the first prototype in terms of its strengths, weaknesses, and risks; (2) defining the requirements of the second prototype; (3) planning and designing the second prototype; (4) constructing and testing the second prototype.

4. At the customer's option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
5. The existing prototype is evaluated in the same manner as was the previous prototype, and, if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
6. The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
7. The final system is constructed, based on the refined prototype.
8. The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.²

The Prototyping Model is a model in which the system is built, tested, and then reworked as necessary until a suitable prototype is at last achieved. By doing this the final system or program can finally be developed. This type of development is used for businesses where the requirements of the system needed are not known ahead of time. Many companies are not sure on the system that is needed to perform the tasks which they are conducting in their business. "It is an iterative, trial-and-error process that takes place between the developers and the users"(Techtaget). As the users figure out what they are looking for as the process goes further and further down the line they realize what their needs are throughout

² Techtaget

the time of development. There are some disadvantages of this system though one being that the user of this system will not be able to use the system for some time because a prototype is being made not the final product due to the uncertainty of the companies goals of the system design. The designers and the users can often focus too much of their efforts on the interface design rather than producing the system which the business will be using. The steps of the model consist of the following: First, the new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system. This leads to a preliminary design that creates for the new system. A first prototype of the new system is constructed from the preliminary design. This usually a scaled-down system, and represents an approximation of the characteristics of the final product. Soon after the users thoroughly evaluate the first prototype, noting its strengths and weaknesses, what needs to be added, and what should be removed. The developer collects and analyzes the remarks from the users. The first prototype is being modified and based on the comments supplied by the users, and a second prototype of the new system is constructed. The second prototype is evaluated in the same manner as was the first prototype. The preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired. The final system is constructed, based on the final prototype. The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime (target).

Exploratory Model, this model is self explanatory in which the name says it all. The process of this type of system development is on of trying a bunch of different designs until one of them seems to be the correct one to develop. Although this process is like the prototyping model it serves it's purpose in a less formal approach. This system is for the business that doesn't know what they want and they need many options to choose from. This process is based on the guess and check method to figure out what is the best one for the task at hand. The problem of this process is the time required to produce a final product. If there is a deadline for the company then this process is not the right one for them. On the contrary this method will allow the designers to produce the perfect system for your company if there is no specific deadline. The steps of this process are as follows:

1. A starting point is determined for the work. All the information available is gathered together in an attempt to get an idea of what the new system will be expected to do, and how it can be done.
2. A rudimentary first-generation system is put together, based on the information gathered and the ideas formulated in the first step.
3. The first-generation system is tested to see how it performs, what it can and cannot do, and what might be done to improve it.
4. A second-generation system is developed from the first one, based on the improvements proposed in the previous step.
5. The second-generation system is tested, as was the first. Its performance is evaluated, and possible improvements determined.

6. The process is repeated as many times as necessary to obtain user satisfaction, or until it is decided that the project is unworkable.
7. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime (techtarget).

The Rapid Application Development is the process takes the method of producing the final system faster and of more quality. The methods that this system uses for the faster production of a system start with the gathering of requirements by way of focus groups and meetings. There is a less formality use of production which allows the designers the freedom to produce at a rapid pace with a more rigid schedule of production. This model is ideal for the companies who need there system as soon as possible. There also is an emphasis in user friendly development or simplicity meaning that the system is generally easier to use. The problem with this system is the reduced scalability meaning there are usually less features in the system produced.

Joint application development is a process involves the client or user of the system being developed as a mediator through out the entire process. The user collaborates with the designer through out the process which gives them the ability to give insight on what they are looking for in the system being developed. This is great for the user by giving them the on site privilege of being involved in the process which normally they wouldn't be able to be involved in. This process gives more client satisfaction and faster development time by which the client has more of a personal ownership of the software developed. The downfall to this process is that it creates more possibilities for conflict between the user and the designer.

Extreme programming of XP is an approach which emphasizes the results of the business first, and uses a “get-something-started” approach for the development of the product using the technique of continual testing and revision. The main idea of this process is to reduce the cost of change. By implementing the concept it should produce a more flexible reason for change. By obtaining feedback and using that feedback to gain knowledge the process is very productive and successful in the regards of keeping it as simple as possible. The idea of starting simple and keeping the design convenient rather than complicated will bring future production faster and more efficient. Although the process is simple many companies like using it for the reason of its simplicity and easy access. “Rather than specialize, all team members write code, test, analyze, design, and continually integrate code as the project develops”(techtarget).

The Agile development method is one that focuses on flexibility rather than just a black and white view of a system. The focus of this is to keep the code simple and flexible for change. By testing often and delivering the new segments when they are completed it allows for great innovation and consistency within the system. “The goal of ASD is to build upon small client-approved parts as the project progresses, as opposed to delivering one large application at the end of the project”(techtarget). The main reason why this system is so effective is that it is adaptive and ready to change for any reason. There are though many criticisms which include: lack of structure and necessary documentation, only works with senior-level developers, incorporates insufficient software design, requires too much cultural change to adopt and can lead to more difficult contractual negotiations.

There are many advantages of SDLC. For businesses that need a system for the first time, to businesses that already have a system and just need to upgrade to meet their new requirements. We are going to focus the most beneficial advantages that help companies. The first advantage is the systematic approach; it has the standardized steps that need to be followed. Formal documents and formal procedures for proceeding from one step to the next are helpful for keeping good track of what is going on. The written evidence provided can be used to arbitrate disputes between developers and users; this is good because you can find out who caused an error while developing the system. It is also very appropriate for transaction processing system(TPS), which is used to collect, store, modify, and retrieve the transactions of an organization. SDLC is also good because it addresses both business and technical issues. Users get to speak up in what they like about the system, what they don't like, and what need to be changed. For the technical issues, the maintaining step of developing the system is what takes care of that.

Now that you saw some of the advantages SDLC has to offer, you need to look at the disadvantages of the system. There are also many disadvantages, but we will only look at the most common ones. The biggest problem there is with SDLC is the budget. Most of the projects end up going over budget costing the company more than what they originally wanted to spend in setting up the system. A second disadvantage is that very little is known on how long the whole process is going to take, and that is why the company ends up under estimating the budget. Another problem in developing a system is that it's hard to know the well requirements that concern a particular feature, so the developer may create something that was not needed or if there is too much unknown for a technique to work well. The requirement analysis may lead to requirement paralysis

because the requirements could have not been explained well. Also the next common problem is that since there is too much to document and it is time taking and costly, proper documentation of all the steps can be neglected leading to future problems. All of those disadvantages can lead to the greatest disadvantage, which is that the user requirements are not being met by the system developed and all the time and money spend is a waste.

After a company knows all the advantages and disadvantages of using the SDLC, and they decide to go ahead and develop a system, they will be contributing to the \$275 billion spend each year in software development projects. They will make many mistakes while developing the project. Research shows that 80% of the projects fail for various reasons, and 30% of them are cancelled before completion because of poor execution. According to DR. Michael Stovsky, “the most common project problems are due to failure to manage project elements successfully.”³ The problems that are encountered along the way are that the requirements are not clearly defined, and agreed upon by both parties. The resources are not adequately planned and allocated causing the project to come to a stop half way. Threats to project success are not clearly detected, identified, and protected against. The project progress is not tracked adequately; leading to too little data collected of data is ignored or poorly understood.

We have some testimonials about companies that have used SDLC in their company. The first testimonial explains their satisfaction of using SDLC. The second testimonial also talks about how they are happy about their SDLC class including smaller group activities and ensure congruence with their general SPS project. Lastly, the third

³ Quick Study

testimonial, an Enterprise Project company inform how pleased they are with the new SDLC methodology. It helped the improvement of their corporate IT department.

"There is a direct correlation between the quality of the individuals working on information technology projects and the time it takes to develop and implement a new system. To date, you have provided us with extremely high-quality individuals across the board. Your consultants have proven to possess superior business analysis and technical skills. We appreciate the fine job you and your firm have done for us."

-J.N.

MIS Director

"You significantly raised the awareness and competency of Project Management in one of our large divisions. On a separate assignment, your consultant revised our Systems Development Life Cycle (SDLC) Training Class to include small group activities and ensure congruence with the general SPS Project Management approach. Please accept our acknowledgement of the skills and professionalism of your staff."

-S.D

Director, Information Technology

"Christine drove standardizing the Systems Development Lifecycle (SDLC) methodology for our corporate IT department. She created a style guide

that was embraced by the company, along with the associated SDLC templates. Her output was detail-oriented and while she worked quite independently, she didn't hesitate to clarify issues that required our attention. Christine jumped into an active project, quickly grasped its goals and rapidly delivered for us."

- Laurie Doolittle, Sr. Manager, Enterprise Projects, Matson
Navigation Company, Oakland

After studying the SDLC, we have come up with some recommendations for companies who will be interested in develop a new information system. These recommendations will help a company accomplish their objective goals in the organizations, such as completing projects on time, budget strategies, and developing systems that will benefit the company. The recommendations are from a variety of industry sources.

- Make sure that you've chosen a systems development life cycle model that suits your project, because every one of the processes involved depends on the model. That said, however, implementing any model is better than none -- much of the success of a project depends upon how scrupulously the model is adhered to.
- Reuse software components when it's appropriate, but don't use code that doesn't work perfectly for its intended purpose just because you have it on hand.
- Be very thorough in gathering requirements, ensuring that all parties agree on what they are -- and make sure you document them.
- Don't promise the moon, if you can't deliver it. Avoid letting someone who isn't fully informed negotiate with the client.

- Make sure that the architecture you've chosen is appropriate for the application you're building. To retain perspective, you might want to look at the smaller picture and plan the architecture incrementally.
- Change is part of life, including software development. You have to accept that various things -- requirements, for example -- are likely to change throughout the life of the project. Keep control of them, but not too rigidly.
- Set up peer review processes for every element of the project.
- Design thoroughly and with care, but remember the KISS principle: Keep it simple.
- In your project plan, split big projects into manageable chunks, with concrete milestones and deadlines.
- Ensure accountability: make sure that deadlines are clear and that people have to report on whether they made them, and explain why not if they don't.
- Implement quality control procedures throughout the project.
- Test exhaustively -- there's no point in doing a cursory run through only to have an application fail when you run it for the client.
- After the project is completed, conduct a thorough post-mortem, with everyone involved. You'll see what worked well and what should have been differently, and your future projects will benefit.

The followings are common questions that companies who are involved with SDLC.

1. We have already tried a life cycle and it did not work, why is it different this time?

The life cycle you currently have was designed some time back and the life cycle can

be outdated. The current one does not reflect all the advantages in engineering. This new one will pay more attention to the training, guidance, and tools. But remember, success will depend on the level of commitment of everybody involved.

2. There is too many tasks and documentation. Is it really necessary to do all these?

The life cycle provides the tasks which must be considered for your project. . All the steps must be documented in order to keep a good record for future reference.

3. How are you guys teaching us if you are still learning the life cycle?

Your focus will not be what we know, but what you can learn to build a system to meet your business needs. First you must learn the life cycle, then if you need help we can get outside sources.

4. If I am already doing most of these things, why should I learn the life cycle?

If you are already doing many of these things, it will be easier to learn the life cycle. Another reason is that if you are doing this your own way, it makes it harder to work in coherent teams. The life cycle combined with what you already know will make you have a bigger advantage over your competitor.

Conclusion

In conclusion, SDLC system involves many phases and steps with different versions. SDLC improves many companies in different ways as in improving the management services, help to budget the company, inventory and systematize organizations. However, there are advantages and disadvantages in the SDLC systems. Understanding and learning these differences will help in the process of considering using a SDLC system. Whether or not you or your company will ever becomes interested in using these systems; the knowledge of there existence and the capabilities that they offer will be a asset ones awareness.

SOURCES

1. System Development life cycle: Objectives and requirements

<<http://www.benderrbt.com/Bender-SDLC.pdf>>

2. Learn IT: Software development –Whatis.com

<http://whatis.techtarget.com/definition/0,,sid9_gci936454,00.html>

3. Quick Study: System Development Life Cycle

<<http://www.computerworld.com/developmenttopics/development/story/0,10801,71151,00.html>>

4. IS auditing guidance. 5 April 2003. System Development Life Cycle (SDLC) Reviews

<<http://www.isaca.org/ContentManagement/ContentDisplay.cfm?ContentID=26905>>

5. Wikipedia. 29 May, 2007.

<http://en.wikipedia.org/wiki/Agile_software_development>

6. SearchSoftwareQuality.com Definitions. 15 Feb 2007. WhatIs.com.

<<http://searchsoftwarequality.techtarget.com>>

SDLC

“System Development Life-Cycle”

Hang Pham

Jared Jelacich

Hector Arreola

HTM 304

Tues/Thurs

