Assignment 1

|  |  |
| --- | --- |
| to: | Joey Kitson |
| from: | Aaron steele |
| subject: | CIS-2245 System Analysis and Design |
| date: | 2021-09-04 |
| CC: | Joey Kitson |

1. **Describe (in your words) what an SDLC is and explain each of the CORE processes (phases within).**

SDLC stands for System Development Life Cycle. A SDLC helps identify all activities needed to research, build, deploy, and maintain information systems. There are 6 core processes and 4 main phases that SDLC approaches have. The following 6 processes of SDLC are Identify, Plan/Monitor, Discover, Design, Build, and Complete. The following 4 phases of SDLC are Inception, Elaboration, Construction, and Transition.

**Below is a breakdown of the 6 core processes:**

**Identify:** The developers must see that there is a problem, identify what exactly is the problem, and get approval to continue the project. Often developers will create rough cost estimates, along with rough risk, cost, and rewards estimates to see if the project is worth continuing for the client.

**Plan/Monitor:** The developers then elaborate on the project on what exactly needs to be done, how might they go about doing that task, along who might be best suited for the task. The developers should begin to monitor the project to ensure that the project goals are being met.

**Discover:** The developers must gather information and gain a better understanding of the details of the problem along with gathering information on how they might go about implementing tasks to solve the problem.

**Design:** The developers then begin to flesh out the details about how tasks should be done along with detailing how their solution will function.

**Build:** The developers then begin to construct and integrate their solutions using information gathered from the previous processes as a guideline.

**Complete:** The developers are finalizing their solutions, testing to see if their solution has met their client’s needs.

**Below is a breakdown of the 4 phases:**

**Inception:** The following are the objectives of the Inception phase create a rough version of the project’s vision, define the scope of the project, create the business case, generate a rough cost estimate of the project, define a rough schedule, and obtain approval. This phase is typically done in one iteration.

**Elaboration:** This phase builds on the Inception phase further defining the project’s vision, identifying the requirements of the project, refining the scope, resolving high risks, and creating more accurate cost estimates.

**Construction:** This is the phase where the project is constructed using the information, planning, and guidelines set by the Inception and Elaboration phases.

**Transition:** In this phase the project is near or at completion, there should also be final testing and inquiries regarding the satisfaction of the client.

**2. Compare and contrast the differences between a prescribed (structured) SDLC versus a more adaptive SDLC. Give an example of each.**

A prescribed (structured) SDLC is otherwise known as the predictive approach to SDLC. In a predictive approach, the team and members should fully understand the problems and complexities of creating or modifying a system. The predictive approach is best used for building systems that are well understood and defined. A big drawback to the predictive approach is that it struggles with being flexible and typically does not do well with change or unforeseen challenges. A good example of a prescribed (structured) SDLC is the waterfall method where the project goes through the 6 core processes but much like a waterfall once a phase was complete there was no going back. Prescribed (structured) SDLC were among the first types of SDLCs being prevalent roughly from the 1970s to the 1990s until more adaptive SDLC were developed.

Unlike A prescribed (structured) SDLC An adaptive SDLC neither the team nor the members fully understand the complexities of creating or modifying a system. Instead, an adaptive SDLC focuses more on adapting to change rather than following a strict guideline or plan. A good example of an adaptive SDLC is the walking skeleton development or the incremental development where multiple iterations are used. The part that makes an adaptive SDLC so flexible is its use of iterative development.

1. **What is Agile development?**

The textbook describes Agile development as the guiding philosophy and set of guidelines for developing Information systems in a rapidly changing environment. There are 4 core philosophies that Agile development contains, below are the following philosophies:

* Respond to change over the plan.
* Individuals and interactions are more important than processes and tools.
* Working software is more valuable than comprehensive documentation.
* Customer collaboration is more valuable than contract negotiation.

Each Agile model follows a set of modeling principles. Below are the following principles for each agile model:

* Develop software as your primary goal.
* Enable the next effort as your secondary goal.
* Minimize your modeling activity (few and simple).
* Embrace change and change in increments.
* Model with Purpose.
* Build multiple models.
* Focus on Content instead of representation.
* Learn from each other and open communication.
* Understand your models.
* Adapt to specific project needs.
* Maximize stakeholder ROI.

Agile SDLC uses Iterative development makes because in iterative development the project repeats itself developing through each iteration allowing for changes to be made in future iterations like the walking skeleton model. Most Iterative development models go through 6 iterations, one for each of the core processes. Below are the details of each iteration and what they entail, note the not every model goes through 6 steps some models such as the spiral model only go through 4.

The first iteration primarily focuses on identifying the problem and planning, there is some discovery in the first iteration, a minor amount of designing and building, and nothing towards completion.

The second iteration primarily focused on the discovery, with all the other core processes being secondary and nothing towards completion.

The Third iteration mainly focuses on Building and testing with discovery and design being secondary. This is the only iteration that has all amounts of the 6 core processes although minor. This is the first iteration that touches on completion.

The Forth iteration is very similar to the Third iteration except there is more focus on building and completion, and no focus on identifying the problem.

The Fifth Iteration Is much like the Forth Iteration, with the fifth still primarily focused on building, the main difference being that there's more focus on completion.

The Sixth and final Iteration mainly focuses on completion, and building secondary, with a very minor amount focused on planning. The Sixth Iteration has no focus on identifying, discovery, or design.