Modern Systems Analysis and Design, Chapter 1 (pp. 1–11, 20–21)

Modern Systems Analysis and Design, Chapter 6 (pp. 144–167, 171–176)

* What does a systems analyst do? Why is it important?
* Why are the different SDLC phases important to systems development?

First let me introduce myself, I’m Dylan and I’m from Denton, Texas. Where currently we are getting a break from consistent 110-degree weather. Even though it’s 99, I’ll take it at this point. I have a lot of interests, from playing guitar, reading sci-fi and sword and sorcery novels like Dune or Conan the Cimmerian. For my family I have a girlfriend and a 1-and-a-half-year-old daughter named Sofia. As for pets, I have three cats, two black cats and a Siamese. They will certainly let you know they’re hungry starting at 5am.

My favorite course in this program so far has been the Computer Languages course, simply because it was my first introduction into the C++ language. Which I found easier than Java at the time, but both are equally as important for object-oriented programming.

The major phases of the systems development lifecycle (SDLC) are the Planning, Analysis, Logical Design & Physical Design, Implementation, and Maintenance. The first phase Planning is where requirement needs are determined either for a client or an organization. Once the needs are determined, then the needs are further prioritized and translated into an actionable plan. This also goes for client requests too, depending on what the client outlines for their requirements.

The second phase, analysis, is where an analyst will come in and decide which procedures to use to take actionable steps towards meeting the requirements. The analysis phase can also be divided into two subphases, where the first is called the requirements determination which is where the analysts will determine from the users what they want from the new system. The second subphase is where the analysts will study the requirements and then carefully structure them depending on their interrelationships.

The third phase, Design can be split into two phases, the logical design phase, and the physical design phase. Logical design phase is where the focused solely on the requirements of the system that are independent of any hardware. How does the structure fit together conceptually rather than physically. Physical design is the next step, where the analysts will then convert the logical design into a physical one, implementing the conceptualized design and then try to form a working model of the system using necessary hardware.

The fourth phase, implementation, is where the physical design is now tested and used. This is usually where all the coding and repeat testing is done by the programmers. For example, in an agile process the programmers will immediately test the code as it is deployed or implemented so any issues can be fixed immediately and then shifted back to a previous phase if not completed or up to specifications. This phase is crucial because anything that is missed in the implementation phase can lead to problems with the system is fully deployed.

The last phase, maintenance, is where any last modifications are made to the system by users that are working on the system in real time. The users may be able to address problems with the system or code and then suggest a better solution that can be updated in the previous phases.

When considering the various ways that requirements can be collected for a system, the method I’m most familiar with and that works for me is in-person conversation. I think this is a very useful method because it allows you to gain insight from the client or user into what the requirements are in their own words. This helps to alleviate any of the misinterpretations that may occur from reading a document, or joint-application-design. If I was collecting user requirements, I would have them express in their own words during a conversation setting the specific requirements they want met.

Technology has made it easier in my opinion for users to collect requirements. For instance, if you have an eCommerce store and you want to figure out how to increase traffic to your site then you can view similar sites that have similar items and determine what may be better or worse about that site and implement those changes to your site. If a larger organization requires its users to determine what requirements are needed for a new system or site, those users can use the internet to search for similar products or services to implement similar characteristics or know what to avoid. This ultimately makes it easier for users to determine requirements because most of the legwork can be completed digitally as opposed to having to physically determine requirements. P.170

Maintaining an ethical standpoint when collecting requirements is crucial to the success of the system. When considering data collection, it’s important to consider privacy concerns during requirements sourcing. Security is also a big problem, when considering how the data is handled for use of obtaining requirements. Also, in the context of data collection, bias is an ethical concern as well because the information obtained must accurately reflect reality instead of reflecting a particular bias.

Valacich, J. S., & George, J. F. (2019). Modern Systems Analysis and Design (9th ed.). Pearson Education (US). https://mbsdirect.vitalsource.com/books/9780135172827