Systems Development Life Cycle (SDLC)

[Chapter 1 & 2]

System Analysis and Design

-- Kendall and Kendall

-- Elias M. Awad

Prepared by - Mijanur Rahaman,
Asst. Prof. of CSE, BUBT

7/19/2016

Problem analyzing and solving Skills

- **1. Defining the problem.** Studies show that defining the problem and taking action occur almost simultaneously for most people, so the more effort you put on the front end, the easier it is to come up with a good solution. Stop and first define what the problem is and isn’t. Since providing solutions is so easy for everyone, it would be nice if they were offering solutions to the right problem. Be a chess master. Chess masters recognize thousands of patterns of chess pieces. Look for patterns in data; don’t just collect information. Put it in categories that make sense to you. Ask lots of questions. Allot at least 50% of the time to defining the problem.

Problem analyzing and solving Skills

- **2. Results oriented impatience.** The style that chills sound problem solving the most is the results driven, time short and impatient person. He/she does not take the time to define problems and tends to take the first close enough solution that comes along. Studies have shown that on average, the solution somewhere between the second and third one generated is the best. So, discipline yourself to pause for enough time to define the problem better and always think of three solutions before you pick one.
- **3. Watch your biases.** Some people have solutions in search of problems. They have favorite solutions. They have biases. They have universal solutions to most situations. Do honest and open analysis first. One of your solutions may in fact fit, but wait to see if you’re right about the problem.

Problem analyzing and solving Skills

- **4. Get out of your comfort zone.** Many busy people rely too much on solutions from their own history. They rely on what has happened to them in the past. They see sameness in problems that isn't there. Beware of "I have always..." or "Usually I...." Always pause and look under rocks and ask yourself, is this really like the problems I have solved in the past?
- **5. Asking others for input.** Many try to do too much themselves. They don't delegate, listen or ask others for input. Even if you think you have the solution, ask some others for input just to make sure. Set up a competition between two teams, both acting as your advisors. Call a problem-solving meeting and give the group two hours to come up with something that will at least be tried. Find a buddy group in another function or organization that faces the same or a similar problem and both of you experiment.

Problem analyzing and solving Skills

- **6. Perfectionist?** Beware of analysis paralysis. A good rule of thumb is to analyze patterns and causes to come up with alternatives. Many of us just collect data, which numerous studies show increases our confidence but doesn’t increase decision accuracy. Perfectionism is tough to let go of because most people see it as a positive trait for them. Recognize your perfectionism for what it might be – collecting more information than others do to improve your confidence in making a fault-free decision and thereby avoiding risk and criticism.
- **7. Incrementalism.** Sometimes the key to bigger problem-solving is to make them into a series of smaller problems. People who are good at this are incrementalists. They make a series of smaller decisions, get instant feedback, correct the course, get a little more data, move forward a little more, until the bigger problem is under control. Learn to break down problems into pieces and parts and solve them one at a time.

Problem analyzing and solving Skills

- **8. Learn some more problem-solving skills.** There are many different ways to think through and solve a problem.
- ✓ *Ask more questions.* In one study of problem solving, seven percent of comments were questions and about half were answers. We jump to solutions based on what has worked in the past.
- ✓ *To get fresh ideas, don’t speedboat, look deeply instead.* Tackle the most vexing problem of your job – carve out 20% of your time – study it deeply, talk with others, look for parallels in other organizations and in remote areas totally outside your field.
- ✓ *Complex problems are hard to visualize.* They tend to be either oversimplified or too complex to solve unless they are put in a visual format. Cut the problem up into its component pieces. Examine the pieces to see if a different order would help, or how you could combine three pieces into one.

Problem analyzing and solving Skills

- ✓ *Pictorial chart.* It is a storyboard where a problem is illustrated by its components being depicted as pictures.
- ✓ *Sometimes going to extremes helps.* Adding every condition, every worse case you can think of sometimes will suggest a different solution. Taking the present state of affairs and projecting into the future may indicate how and where the system will break down.
- ✓ *Are you or others avoiding making the tough points?* In almost any group, there are topics so hot they can't be mentioned, let alone discussed. Everyone takes three index cards and writes down three undiscussables. The cards are then shuffled and each person receives a different three back. The cards are read, charted, and themes are arrayed for discussion.

Problem analyzing and solving Skills

- **9. Avoiding risks?** Develop a philosophical stance toward mistakes and failures in problem solving. After all, most innovations fail, most proposals fail, most change efforts fail, and the initial solutions to complex problems do not work. The best tack when a solution doesn’t work is to say, “What can we learn from this?” and move on. The more tries, the more feedback and the more chances to find the best answer.
- **10. Disorganized?** Problem solving involves using rigorous logic and disciplined methods. It involves going through checklists, looking under rocks, and probing all fruitful sources for answers. If you’re disorganized, you need to set tight priorities. Focus on the mission-critical few. Don’t get diverted by trivia.

Technical Skills (1)

- Many aspects of your job as a system analyst are technically oriented.
- The following activities will help you stay up-to-date:
  - Read trade publications
  - Join professional societies
  - Attend classes or teach at a local college
  - Attend many courses or training sessions offered by your organizations
  - Attend professional conferences, seminars, or trade shows
  - Participate in electronic bulletin, new groups

Technical Skills (2)

- You should be familiar as possible with information technology:
  - Microcomputer, micro station, workstation, mainframe computers
  - Programming languages
  - Operating systems
  - Database and file management systems
  - Data communication standards
  - Software for local and wide networks
  - Web developing tools
  - Decision support system generators
  - Data analysis tools
  - Data design tools

Management Skills

- System analysts are almost always members of project teams and are frequently asked to lead team.
- Management skills are very useful for anyone in a leadership role.
- There are four class of management skills:
  - 1- Resources
  - 2- Project
  - 3- Risk
  - 4- Change management

Interpersonal skills

- Communication skills
- Interviewing, Listening, and questionnaires
- Written and oral presentations
  - Meeting agenda
  - Meeting minutes
  - Interview summaries
  - Requests for proposal from contractors and vendors
- Working alone and with a team
- Facilitating groups
- Managing exceptions

Important system concepts

- There are several other system concepts with which systems analysts need to become familiar:
  - Decomposition
  - Modularity
  - Coupling
  - Cohesion

Decomposition

- Definition: The process of breaking down a system into smaller component
  - The purpose of decomposition is to allow the system analysts to:
    - Break a system into small, manageable subsystem
    - Focus on one are at a time
  - Concentrate one component pertinent to one group of users
  - Build different components at independent times

Modularity, Coupling and Cohesion

- Modularity
  - Dividing a system up into chunks or modules of a relatively uniform size. To Simplify the redesign and rebuild process
- Coupling
  - The extend to which subsystems depend on each other.
  - Subsystem should be independent as possible. If one subsystem fails and other subsystem are highly dependent on it, then the other will either fail themselves or have problems functioning
- Cohesion
  - A cohesion is the extent to which a subsystem performs a single function.

CASE Tools

- CASE tools are automated, microcomputer-based software packages for systems analysis and design
- Four reasons for using CASE tools are:
  - To increase analyst productivity
  - Facilitate communication among analysts and users
  - Providing continuity between life cycle phases
  - To assess the impact of maintenance

CASE Tool Categories

- CASE tools may be divided into several categories
  - Upper CASE (also called front-end CASE) tools, used to perform analysis and design
  - Lower CASE (also called back-end CASE). These tools generate computer language source code from CASE design
  - Integrated CASE, performing both upper and lower CASE functions

Upper CASE

- Upper CASE tools
  - Create and modify the system design
  - Store data in a project repository
  - The repository is a collection of records, elements, diagrams, screens, reports, and other project information
  - These CASE tools model organizational requirements and define system boundaries

Lower CASE

- Lower CASE tools generate computer source code from the CASE design
- Source code may usually be generated in several languages

Reverse Engineering

- Reverse engineering is generating the CASE design from computer program code
- Source code is examined, analyzed, and converted into repository entities

Reverse Engineering

- Reverse engineering produces (depending on the tool set used)
  - Data structures and elements, describing the files, records, and field
  - Screen designs, if the program is online
  - Report layouts for batch programs
  - A structure chart showing the hierarchy of the modules in the program
  - Database design and relationships

Advantages of Reverse Engineering

- It has the following advantages:
  - – Reduced system maintenance time
  - – Program documentation is produced for loosely documented programs
  - – Structured programs may be generated from unstructured, older programs
  - – Future system maintenance is easier to implement
  - – Unused portions of programs may be eliminated

Assignment

- Describe your university or college as a system.
  - What is the input?
  - What is output?
  - What is the boundary?
  - What is the components and their relationship?
  - The constraint
  - The environment
- Draw a diagram of this system

Assignment

- Describe yourself in terms of your abilities at each of the following interpersonal skills: working alone verse working with a team, interviewing, listening, writing, presenting, facilitating a group, and margining expectations. Where are your strengths and weakness? Why? What can you do to capitalize on your strengths and strengths areas where you are weak?