

Chapter 1: The context of systems analysis and design methods and outline

1. Who are the system stakeholders?

System stakeholders are anyone who has a vested interest in the information system, anyone from the systems user, systems designer, system builder and the systems analyst.

2. What business relationship do the systems analysts have with each of the stakeholders?

The systems analyst is a problem solver and facilitator among the different stakeholders. The SA takes the requirements from each stakeholder, analyses it and proposes a solution. For example, the systems analyst serves as a facilitator of communications between the non-technical system folks and technical system builders and designers.

3. Where do systems analysts work?

Systems Analysts can work within departments supporting each of the business units of a firm, they can work in the IT department or with executive management.

4. What are the roles of systems analysts?

The systems analyst is a problem solver. They can be considered change agents because every new system literally changes the business providing new opportunities, new markets, and new data. Their role is to anticipate problems, solve them, and prevent them. They must also find ways to improve systems even when they are performing, as they should. They must give directives on change

5. What are the career paths for systems analysts?

- Project Managers
- IS Managers
- Technical Specialists (Database, Telecommunications, etc.)
- Consultants

6. What skills/knowledge/attributes do successful system analysts have?

Analytical thinking and problem solving:

1. Creative thinking: thinking creatively and helping others to apply creative thinking helps BA to be effective in generating new ideas and alternatives to problem solving and opportunities
2. Decision making: BA must be effective in understanding the criteria involved in making a decision, and in assisting others to make better decisions.
3. Learning: the ability to quickly absorb new and different types of information allows BA to work effectively in evolving environments

Behavioural Characteristics

1. Ethics: Behaving ethically and thinking of ethical impacts on others allows BA to earn the respect of the stakeholders. (don't share information given with others)
2. Organization and Time Management: skills help BA perform tasks effectively and use work time efficiently
3. Adaptability: BA adjust their behavioural style and method of approach to increase their effectiveness when interacting with different stakeholders, orgs, and situations (look at how they're dressed)

Business Knowledge

1. Industry Knowledge: provides the BA with an understanding of current practices and activities within an industry and similar processes across industries.
2. Organization Knowledge: provides an understanding of the management structure and business architecture of the enterprise

Communication Skills

1. Verbal Communication Skills: used to convey ideas, concepts, facts and opinions to a variety of stakeholders.
2. Non-Verbal Communication Skills: enable the effective sending and receiving of messages through – but not limited to – body movement, posture, facial expressions, gestures and eye contact
3. Written Communication Skills: used to convey ideas, concepts, facts, and opinions to variety of stakeholders
4. Listening: allows the BA to accurately understand information that is communicated verbally (active listening: follow up questions, ask for further information)

Interaction Skills

1. Teamwork: skills allow BA to work productively with team members, stakeholders and any other vested partners so that solutions can be effectively developed and implemented.
2. Teaching: skills help BA to effectively communicate BA info, concepts, ideas and issues.

7. What are the business drivers for today's IS and how is each relevant to systems analysis?

- Globalization of the economy: the IS and any applications must be internationalized. They must support multiple languages, different time zones, different trade regulations, currency exchange rates and different business and cultures. When consolidating different business IS around the world, the ability to speak the native language/slang can help reduce language barriers.
- E-Commerce and Business: B2C (research, order and pay for products) and B2B (procurement of goods, shipping, ordering, payment)
- Security and Privacy: Security and privacy controls. Regulations from one country to the next.
- Collaboration and Partnership: Wal-Mart for example can integrate their systems to their suppliers, so every time the supply dips, they can automatically reorder.
- Knowledge Asset Management: A lot of data coming in, firms want to be able for their systems to manage it, store it properly and then integrate it.
- Continuous Improvement and TQM: Systems analysts are part of the TQM chain
- Business Process Redesign: Build your process or buy it. Write your own code to build it or buy an off the shelf product that will most likely require you to modify your process a bit to integrate the software. Maximum efficiency and lowest possible costs.

8. What are the technical drivers for today's IS and how is each relevant to systems analysis?

- Networks and the Internet: XML, Intranet, Extranet
- Mobile and Wireless Technologies: Blackberry, iPhone
- Object Technologies: Agile, Object-oriented analysis and design
- Collaborative Technologies: Email, IM, Skype
- Enterprise Applications: Supply Chain Management

9. What is Business Analysis according to BABOK Guide?

- Business analysis is a set of tasks and techniques that are used to work as a liaison among stakeholders to understand the structure, policies and operations of an organization. It is also used to recommend solutions to the organization.

- Business analysis involves understanding the organization as a whole, how they operate, defining their capabilities as an organization to deliver to their external stakeholders. It is understanding their objective and how the goals connect towards that objective.
- Business analysis can be performed to understand the current state of an organization or to serve as a basis for later identification of business needs.
- The Business is the “translator” they take all the information from different stakeholders, mainly their needs, not their wants and then synthesize the information among organizational units.

System Concepts

1. What is a system?

- Organized, interrelated, and interacting entities or ideas that work together to achieve a common goal
- They exist in an environment
- They can be divided into smaller systems

2. What is systems approach?

The system approach is the way of thinking about the job of managing. It provides a framework for visualizing internal and external environmental factors as an integrated whole. It allows recognition of the function of subsystems, as well as the complex, suprasystems within which orgs must operate.

3. What are the components of systems?

1. Boundary
2. Entities
3. Subsystem:
4. Input
5. Output
6. Interface
7. Buffers
8. Feedback and Control
9. Constraints

4. Represent systems' components in a DFD

- Entities and subsystems are represented by process outline
- Buffers are represented by data store outline
- Inputs/outputs are represented by data flow outline
- Entities in the environment are represented by external entity outline

Modelling System Requirements with Use Cases

1. Know how to identify actors

Use cases are written from the POV of the actor

- ▶ An **actor** represents an external entity that needs to **interact with the system** to exchange information. An actor is a user, a role, which could be an external system as well as a person.

Questions to ask when looking for actors

- Who or what provides inputs to the system?
- Who or what receives outputs from the system?
- Are interfaces required to other systems?
- Are there any events that are automatically triggered at a predetermined time?

- Who will maintain information in the system?
- Does anyone provide a service to the system?

2. Know how to identify use cases

- ▶ A use case is a complete sequence of related actions (a scenario), both automated and manual, for the purpose of completing a business function: What the system must do.

Questions to ask when looking for actors

- What are the main tasks for the system?
- What information does the actor need from the system?
- What information does the actor provide to the system?
- Does the system need to inform the actor of any changes or events that have occurred?
- Does the actor need to inform the system of any changes or events that have occurred?

3. Know how to develop a use case model given a narrative description of a system

- Identify actors (look at the sources and destinations of major inputs and outputs)
- Identify use cases (major system functions)
- Identify the system boundary
- Identify associations between actors and use cases
- Identify additional associations between use cases (“extend”, “include”)
- Identify inheritance relationships among use cases and actors

Chapter 3: Information Systems Development and Outline

1. What is CMM? Distinguish between the levels of CMM.

The Capability Maturity Model (CMM) is a methodology used to develop and refine an organization's software development process. The model describes a five-level evolutionary path of increasingly organized and systematically more mature processes

CMM Levels:

Level 1—Initial: System development projects follow no prescribed process.

Level 2—Repeatable: Project management processes and practices are established to track project costs, schedules, and functionality.

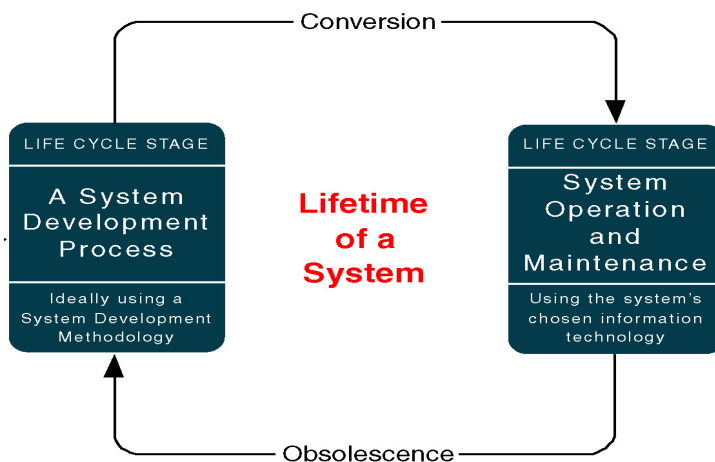
Level 3—Defined: A standard system development process (a “methodology”) is purchased or developed. All projects use a version of this process to develop and maintain information systems and software.

Level 4—Managed: Measurable goals for quality and productivity are established.

Level 5—Optimizing: The standardized system development process is continuously monitored and improved based on measures and data analysis established in Level 4.

2. What is SDLC?

The factoring of a lifetime of an IS into two stages, (1) systems development and (2) systems operation and maintenance – first you build it; then you use and maintain it. Eventually, you cycle back to redevelopment of a new system.



3. What are the principles underlying systems development?

1. Get the owners and users involved in all system development phases.

- User Participation/Involvement creates “System Ownership” and leads to User Acceptance and User Satisfaction.

2. Use a problem solving approach

- Study and understand the problem in its context
- Identify candidate solutions and select the best available

3. Establish phases and activities (define a process to follow)

- Scope definition
- Problem Analysis
- Requirement Analysis
- Logical Design

- Decision Analysis
- Physical Design and Integration
- Construction and Testing
- Implementation and Delivery

These phases identify problems, evaluate, design, and implement solution (Systems Development Process)

4. Document throughout the system development process

- Ongoing activity to reveal strength and weakness of the system during the development process
- Enhance communication and acceptance among stakeholders
- Agreements and Contracts between Owner/User and Analyst/Designer on the Scope, Requirements, Resources of the project.

5. Establish standards to ensure system integration involving:

- Database Technology: oracle, Microsoft, SQL
- Software Technology: visual C#, Java
- Interface Technology: how will the user interfaces be developed?

6. Manage the process and projects

- Process management: concerned with phases, activities, deliverables, and quality standards should be consistently applied to all projects.
- Project management: the process of scoping, planning, staffing, organizing, directing, and controlling a project to develop an information system at a minimum cost, within a specified time frame, and with acceptable quality.

7. Justify Systems as Capital Investments

- There are several possible solutions, the first one is not necessary the best

8. Don't be Afraid to Cancel and Revise Scope: Creeping Commitment

- Expectation and scope of a project may be growing up
- Development process has checkpoints for its phases: all costs committed so far are sunk costs.
- Cancel the project if it is no longer feasible (ORGANIZATION)
- Reevaluate/adjust cost/schedule if the scope is expanding (ANALYST)
- Reduce the scope if budget/schedule is shrinking (ANALYST)

9. Divide and Conquer

- Divide a complex system into simpler subsystems/components
- Problem solving process could be simplified for smaller problems
- Different subsystems for different stakeholders

10. Design Systems for Growth and Change

- Changes of technology, user requirements
- Flexibility and adaptability should be built into the system

4. Where do system development projects come from?

Unplanned projects triggered by:

- Problems

- Opportunities
- Directives

Planned projects triggered by:

- Information strategy plan
- Business strategy plan

5. Be able to analyse an info system using the PIECES Framework

6. Which activities overlap much of the SDLC?

7. What are the phases of the SDLC and what tasks do they include?

1. Project Initiation

Scope Definition:

- Purpose: define perceived problems, opportunities, and directives; assess the risk of project; establish scope, preliminary requirements and constraints, participants, budget and schedule (preliminary study)
- Issues: Is the project worthwhile? Define the scope of project
- Deliverable: Project charter/plan
- Feasibility check: Cancel project / Approve to continue / Reduce or expand the scope with budget and schedule modification

2. System Analysis

Problem Analysis:

- Purpose: to study and analyze the existing system from the users' perspectives as they see Data, Processes, and Interfaces
- Issue: Cost/benefits of building new system to solve these problems
- Deliverable: system improvement objectives (business criteria to evaluate the new system)
- Feasibility check: Cancel project / Approve to continue / Reduce or expand the scope with budget and schedule modification

Requirement Analysis:

- Purpose: discover users' needs or expectations out of the new system in terms of Data, Processes, and Interfaces
- Issue: Specify requirements for the new system (WHAT IS TO BE DONE) without prematurely expressing technical details (HOW)
- Errors and omissions in requirement analysis result in user dissatisfaction of final system and costly modifications
- Deliverable: business requirements statement

Logical Design:

- Purpose: translating business user requirements into a system model that depicts only WHAT TO DO without specifying any possible technical design or implementation of those requirements (conceptual design).
- Issue: using graphical model of a system to represent user requirements in terms of Data, Processes and Interfaces, and to facilitate improved communication between system stakeholders.
- Deliverable: Logical Systems Models (DFD, ERD, Use Case Diagrams, Class Diagrams, etc.)

Decision Analysis:

- Purpose: identify all candidate solutions, analyze the feasibility of each candidate, recommend a candidate system as the target solution
- Issue: Feasibility analysis in terms of technical, operational, economic, schedule (TOES), and risk
- Deliverable: approved system proposal
- Feasibility check: Cancel project / Approve system proposal with budget and schedule modification / Reduce the scope of proposed solution with budget and schedule modification

3. System Design

Physical Design:

- Purpose: to transform business requirements into technical design specifications for construction
- Issue: HOW technology will be used to build the system in terms of Data, Processes, and Interfaces
- Design by Specifications vs. Design by Prototyping
- Deliverable: System design specifications
- Feasibility check: Continue/ Reduce or expand the scope with budget and schedule modification

Construction Phase:

- Purpose: to build and test a system that fulfills business requirements and design specs; implement interfaces between new and existing systems
- Issue: Construct database, application programs, user/system interfaces, implement software
- Deliverable: proposed system within budget and schedule

4. System Implementation

Installation and Delivery:

- Purpose: deliver the production system into operation
- Issue: Train users, write manuals, load files, populate database, final test
- Conversion plan: parallel systems, switch point
- Deliverable: system up and running

5. System Operation and Maintenance

- Ongoing system support would be provided until the system becomes obsolete and is replaced by a new one
- Issues: technical support for user, fixing bugs, recovering plan, adapt to emerging requirements
- When a system has reached entropy, new project for new system should be initiated

Summary: Systems Development Process

- ▶ **Scope Definition Phase:** What Business Problem
- ▶ **Problem Analysis Phase:** What System Issues (Info/Data, Processes, Communications/Interfaces)
- ▶ **Requirement Analysis Phase:** What User Needs
- ▶ **Logical Design:** Conceptual Model – What to Do
- ▶ **Decision Analysis Phase:** What Solution
- ▶ **Design Phase:** Physical Model: How to Do
- ▶ **Construction Phase:** Do It
- ▶ **Implementation Phase:** Use It

8. Distinguish between sequential and iterative system development?

Waterfall (sequential) approach

- Each phase must be completed before the next phase can begin and there is no overlapping in the phases.
- The whole process of software development is divided into separate phases
- It allows for departmentalization and control
- Does not allow for much reflection or revision

Iterative approach (incremental & spiral):

- An iterative life cycle model does not attempt to start with a full specification of requirements.
- Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements.
- This process is then repeated, producing a new version of the software at the end of each iteration of the model.
- there is a working model of the system at a very early stage of development which makes it easier to find functional or design flaws
- disadvantage with this SDLC model is that it is applicable only to large and bulky software development projects.

Chapter 6: Requirements Discovery

1. What is the purpose of requirements discovery?

The process and techniques used by systems analysts to identify or extract system problems and solution requirements from the user community.

2. Consequences of failing to correctly identify system requirements.

- The system may cost more than projected.
- The system may be delivered later than promised.
- The system may not meet the users' expectations and that dissatisfaction may cause them not to use it.
- Once in operation, the costs of maintaining and enhancing the system may be excessively high.
- The system may be unreliable and prone to errors and downtime.
- The reputation of the IT staff of the team is tarnished because any failure, regardless of who is at fault, will be perceived as a mistake by the team.

3. Activities in requirements discovery process

1. Problem discovery and analysis (symptoms vs problems)

- Popular tool: Shikawa diagram (fishbone diagram)
Graphical tool used to identify, explore and depict problems and the cause and effect of these problems

2. Requirements discovery

Fact-Finding (Data collection, information gathering):

- Sampling of existing documentation, forms, and databases.
- Research and site visits.
- Observation of the work environment.
- Questionnaires.
- Interviews.

3. Documenting and analysing requirements

Potential Problems in Draft Requirements:

- Missing requirements
- Conflicting requirements
- Infeasible requirements
- Overlapping requirements
- Ambiguous requirements

Requirements Definition Document:

A requirements definition document should consist of the following:

- The functions and services that the system should provide.
- Nonfunctional requirements including the system's features, characteristics, and attributes.
- The constraints that restrict the development of the system or under which the system must operate.
- Information about other systems that the system must interface with.

Requirements Validation

Possible Errors That a System Analyst Might Find During Requirements Validation:

- System models that contain grammatical errors or typographical errors
- Conflicting requirements
- Ambiguous or poorly worded requirements
- Lack of conformance to quality standards required for the document

4. Requirements management to handle changes

Managing Change:

- The process of managing change to the requirements
- How a change request should be submitted
- How it is analysed for impact to scope, schedule and cost

4. Fact-Finding techniques: advantages and disadvantages, do's and don'ts of each

Observation		Questionnaires	
<p>Advantages</p> <ul style="list-style-type: none"> • Data gathered can be very reliable • Can see exactly what is being done in complex tasks • Relatively inexpensive compared with other techniques • Can do work measurements 	<p>Disadvantages</p> <ul style="list-style-type: none"> • People may perform differently when being observed • Work observed may not be representative of normal conditions • Timing can be inconvenient • Interruptions • Some tasks not always performed the same way • May observe wrong way of doing things 	<p>Advantages</p> <ul style="list-style-type: none"> • Often can be answered quickly • People can complete at their convenience • Relatively inexpensive way to gather data from a large number • Allow for anonymity • Responses can be tabulated quickly 	<p>Disadvantages</p> <ul style="list-style-type: none"> • Return rate is often low • No guarantee that an individual will answer all questions • No opportunity to reword or explain misunderstood questions • Cannot observe body language • Difficult to prepare
Interviews		Interviewing Do's and Don'ts	
<p>Advantages</p> <ul style="list-style-type: none"> • Give analyst opportunity to motivate interviewee to respond freely and openly • Allow analyst to probe for more feedback • Permit analyst to adapt or reword questions for each individual • Can observe nonverbal communication 	<p>Disadvantages</p> <ul style="list-style-type: none"> • Time-consuming • Success highly dependent on analyst's human relations skills • May be impractical due to location of interviewees 	<p>Do</p> <ul style="list-style-type: none"> • Dress appropriately • Be courteous • Listen carefully • Maintain control of the interview • Probe • Observe mannerisms and nonverbal communication • Be patient • Keep interviewee at ease • Maintain self-control • Finish on time 	<p>Don't</p> <ul style="list-style-type: none"> • Assume an answer is finished or leading nowhere • Reveal verbal and nonverbal clues • Use jargon • Reveal personal biases • Talk more than listen • Assume anything about the topic or the interviewee • Tape record (take notes instead)

5. Which documents should you collect; which documents should you not collect?

Collect:

Organization chart, studies, minutes, reports that document the problem, accounting records, performance reviews

Not collect

6. What types of questions are used in interviews?

1. Open-ended:

- Generate descriptive answers that encourage discussion
- Require more than a one-word response to promote discussion
 - Do you have any side effects from medications?

2. Closed-ended:

- Generate short, finite answers (i.e., yes or no)
- Do not encourage discussion
 - Do you have symptoms of TB?

3. Probe:

- A question/statement used to gain more information
- Can be open-ended or closed-ended
 - Describe the places you've lived in the past 3 months.

7. What kind of information is sought in an interview?

1. Opinions
2. Goals
3. Feelings
4. Informal Procedures

8. Interviews: how to prepare for them, how to conduct them, what tasks are done post-interview, questions to avoid, etc.

Prepare for the Interview:

- Types of Questions to Avoid
 - Loaded questions
 - Leading questions
 - Biased questions
- Interview Question Guidelines
 - Use clear and concise language.
 - Don't include your opinion as part of the question.
 - Avoid long or complex questions.
 - Avoid threatening questions.
 - Don't use "you" when you mean a group of people.

Conduct the Interview:

- Dress to match interviewee
- Arrive on time
 - Or early if need to confirm room setup
- Open interview by thanking interviewee
- State purpose and length of interview and how data will be used
- Monitor the time
- Ask follow-up questions
 - Probe until you understand
 - Ask about exception conditions ("what if...")

Post interview:

- Compile and categorize data collected
- Compare notes as a team and agree on your understanding

- Prepare follow-up questions
- Write the interview report

9. What is the recommended fact-finding strategy?

Interviews

10. What is “Elicitation and Collaboration” according to BABOK Guide

11. What is “Requirements Lifecycle Management” according to BABOK Guide?

12. What is “Strategy Analysis” according to BABOK Guide?

13. What is “Requirements Analysis and Design Definition” according to BABOK Guide

CHAPTER 9: PROCESS MODELLING and LOGIC DEFINITION

1. Differentiate between logical and physical system models

Logical model is a nontechnical pictorial representation that depicts what a system is and what it does.

Physical model is a technical pictorial representation that depicts what a system is or does and how the system is implemented.

2. DFD Syntax: components of DFD, their labelling and numbering

- Process: transform input to output, used to store data received from external entity to a data store or to send data from a data store to an external entity.
Labelling = imperative phrase with an active verb followed by an object clause (check client's credit)
 - ▶ A **Process** is a **work** or **action** performed on input data flow to produce an output data flow
 - ▶ A process must have **at least one input** data flow and **at least one output** data flow
- External entity: system interacts with it
Label: Singular noun
 - ▶ An **External Entity** is a **provider** (source) or **receiver** (sink) of **data** and info of the system but is **NOT** part of the system
 - ▶ An external entity defines a person, organization unit, or other organization that lies outside of the scope of the system but that interacts with the system being studied
- Data flow: flow of data between processes, data stores and external entities.
Label: descriptive and unique noun.
 - ▶ A **Data Flow** represents a **movement of data** among processes or data stores
 - ▶ A data flow does **NOT** represent a document or a physical good: it represents the **exchange of information** in the document or about the good
 - ▶ A data flow represents an input of data to a process, or the output of data from a process
- Data store: temporary or permanent place to store data
Label: Descriptive unique and plural noun
 - ▶ A **Data Store** is a **storage/inventory of data**: it contains information
 - ▶ A data store is "data at rest" compared to a data flow that is "data in motion"

3. Frequently encountered DFD modelling challengers

4. DFD Guidelines (system level DFD, hierarchies of DFD's, balancing DFD levels, infinite sinks, infinite sources, read-only and write-only data stores, black holes, miracles, grey holes)

Define the system boundaries

Identify the scheduled inputs and outputs

Starting the System level DFD:

- Start from the top left with the external entity
- Continue to the right and then down
- There is no need to number the processes in the system level DFD
- Concentrate on translating all the information gathered
- Once finished evaluate:
 - Black holes (input only processes), grey holes (processes with insufficient inputs) and miracles (output only processes).

- Verify infinite sinks (external entities that only receive output from the system), and infinite sources (external entities that only send input to system).
- Verify the read-only data (data stores that provide only input to processes) and write-only data (data stores that only receive data flows from processes).

5. What is functional decomposition diagram (FDD)? Compare FDD to DFD.

- DFD helps understand the concept problems and operate within the perceived environment: allow us to have a pictorial idea of what the company does.
- FDD is a diagram that partitions the system into logical subsystems an/or functions. Increasingly analysts and users are being asked to ignore organizational boundaries and to build cross-functional systems that streamline processing and data sharing.

6. Know how to represent the logic in a single or group of processes using a decision table

A decision table is a diagram of process logic where all of the possible choices and conditions the choices depend on are represented in a tabular form. In a decision table we have:

- condition stubs - the part that lists the conditions relevant to the decision, (C)
- action stubs - the part that lists the actions that result for a given set of conditions (a)
- rules - the part that specifies which actions are to be followed for a given set of conditions (R)
 - only Y, N or – to represent rules for different conditions

DFD allow us to break down each process until it represents a single task but it does not provide us with tools to represent the if then else logic within a process. A decision table is an example of logic definition tools and supplements DFD.

Correspondence to DFD:

Each condition = data flow going into the process.

Each action = Going out of the process.

7. Develop a DFD given a decision table

8. Know the correspondence between systems terminology and DFD representation