

SDLC - Systems Development Life Cycle

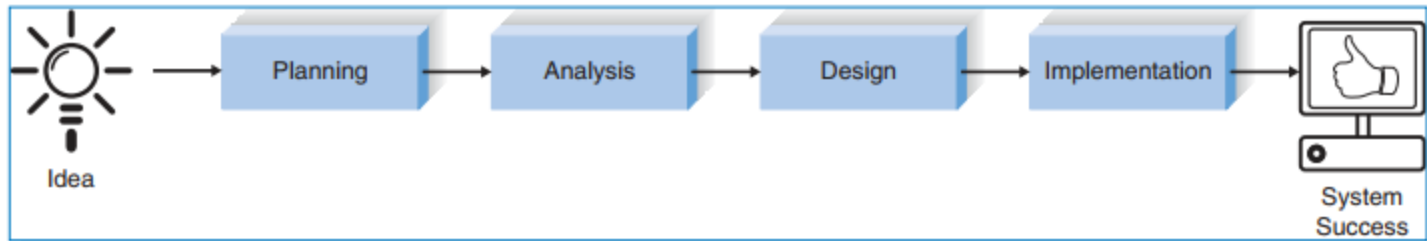
Tugberk Kocatekin
T.C. Istanbul Arel University
Spring 2023

Introduction

- Building an information system is similar to building a house.
- First you have a **vision** of what you want. You hire some people to draw this out for you (**design**) and this goes on until you accept the design.
 - It does not happen just like that. You need to ask your spouse, children, etc. Maybe the design is great but it is very hard to realize.
- Later, **blueprints** are done based on the design. You need to be very careful up to this point because after this, it would be expensive to go back.

Building an information system

- Similar to building a house. There are four fundamental **phases**:
 - Planning, analysis, design and implementation
 - Each phase is composed of a series of steps.



Project identification and initiation

- A project is identified when someone in the organization identifies a *business need* to build a system.
 - Ex: a new marketing campaign, reaching out to a new type of customer, improving interactions with suppliers, etc.
- Sometimes there is a problem in the company and that can be the *need*.
 - Maybe market share dropped.
 - Maybe there are poor customer service
 - Unacceptable product defect rates
 - Increased competition, etc.
- Also possible that organization keep an eye on *emerging technologies* and think about the future. Maybe they want to be the first on market on some technologies.

System Request

- In most organizations, project initiation begins by preparing a *system request*.
- It is a document that describes the business reasons for building a system and the value that the system is expected to provide.
- Most system requests include five elements:
 - Project sponsor, business need, business requirements, business value and special issues.
- For a good example of a System Request, check `System_Request.pdf` from the Lecture Notes.

Example

CONCEPTS

1-C INTERVIEW WITH DON HALLACY, PRESIDENT, TECHNOLOGY SERVICES, SPRINT CORPORATION

IN ACTION

At Sprint, network projects originate from two vantage points—IT and the business units. IT projects usually address infrastructure and support needs. The business-unit projects typically begin after a business need is identified locally, and a business group informally collaborates with IT regarding how a solution can be delivered to meet customer expectations.

Once an idea is developed, a more formal request process begins, and an analysis team is assigned to investigate and validate the opportunity. This team includes members from the user community and IT, and they scope out at a high level what the project will do; create estimates

for technology, training, and development costs; and create a business case. This business case contains the economic value added and the net present value of the project.

Of course, not all projects undergo this rigorous process. The larger projects require more time to be allocated to the analysis team. It is important to remain flexible and not let the process consume the organization. At the beginning of each budgetary year, specific capital expenditures are allocated for operational improvements and maintenance. Moreover, this money is set aside to fund quick projects that deliver immediate value without going through the traditional approval process. *Don Hallacy*

- If you are interested in such matters, please refer to the book: Systems Analysis & Design by Dennis, Wixom, Roth.

Project sponsor

- When a strong business need for an information system is recognized, someone who has interest in the system's success comes forward. It is called a *project sponsor*.
- PS works throughout the SDLC to make sure that the project is moving in the right direction from the perspective of the business.
 - They are generally business people, rarely IT.
- If the project is too large, the project sponsor can be the entire senior management team.

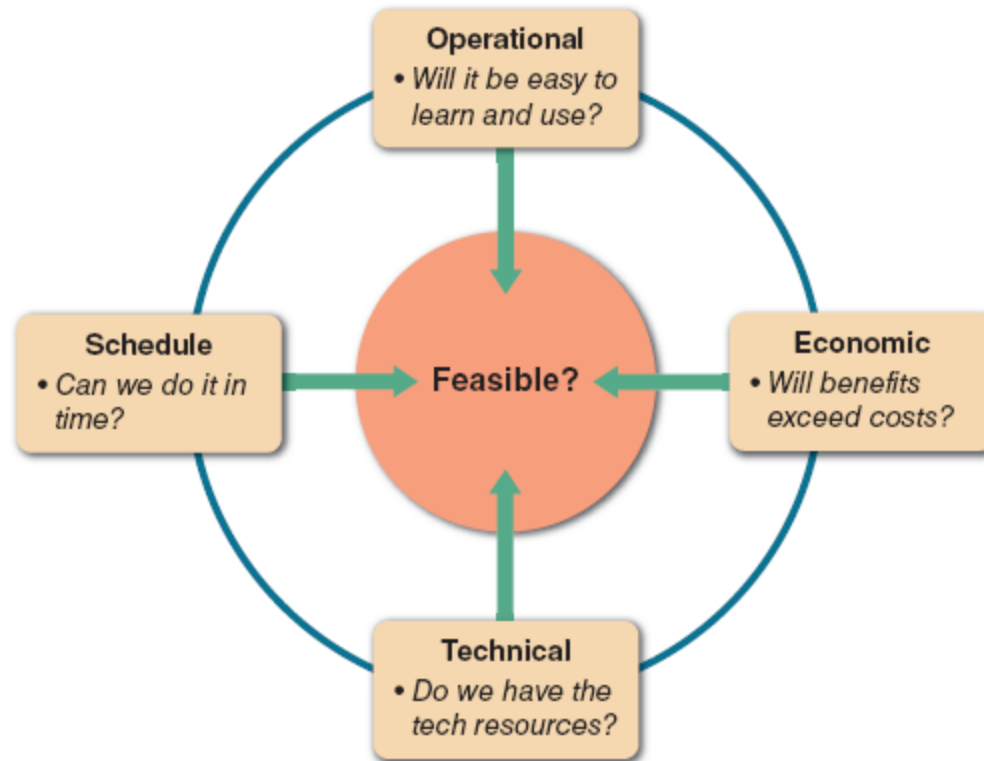
Planning

- In this phase, we try to answer the following questions:
 - Why are we building this system?
 - How the project team will build it?
- First, we try to identify the benefit of this system to the business.
 - Is it going to lower costs?
 - Is it going to increase revenues?
- So, when someone asks for a new *system request*, we do *feasibility analysis* to answer following questions:
 - Can we build it? (Technical feasibility)
 - Will it provide business value? (Economic feas.)
 - If we build it, will it be used? (Organizational feas.)
 - Can we do it in time? (Schedule)

Planning cont.

- After doing the *feasibility analysis* we take this with the system request and present to an committee (called approval or steering committee) which will decide whether the project will be undertaken or not.
- When they **approve**, now it enters *project management*. Here, the project manager creates a work plan, staffs the project and puts techniques to help the project team control and direct the project.

Feasibility



Operational Feasibility

- Does management support the project?
 - Do users support it?
 - Is the current system well liked and effectively used?
 - Do users see the need for change?
- When we build the new system, is there going to be any change in the number of employees?
 - If so, what is going to happen to them
- Will the new system require training?
 - If so, are we prepared for it. Do we have necessary resources for that.
- Will there be changes to users access to information?
 - Maybe the new system will not let the user access to an information.
 - Will performance decline in any way? If so, the overall gain to the organization **must outweigh** individual losses.
- Will it effect the company's image? Is there any ethical issues?...

Technical Feasibility

- Does the company have the necessary hardware, software and network resources?
 - If not, can we buy/rent them? Are we available for it monetary wise?
- Do we need (additional) technical expertise?
- Can the platform evolve for future needs?
- Do we need a *prototype*?
- Will it work with systems of customers and suppliers?
 - Versioning and platforms can be an issue.

Economic Feasibility

- We need to think about the costs.
 - Cost for people, IT staff and users.
 - Cost for hardware and equipment
 - Cost of software: in-house and third party
 - Cost of training
 - Cost of licenses and fees
 - Cost of consultation
 - ..

- We should also remember that not everything has tangible benefits.
(tangible: quantified and measured easily)
 - For example, a user-friendly system can improve employee satisfaction.
 - A better sales system with increased reporting can supply better information for marketing decisions.
 - A new website can enhance the company image, etc.

Schedule Feasibility

- Is there a firm timetable for the project?
- What will happen if we can accelerate schedule?
 - If so, does that mean we can take risks?
- Will the project management techniques be enough for this project?
- Is there a project manager?
- Sometimes, even if a project seem feasible, maybe we don't need it. Feasible does not mean *should be done*.

- In this phase, we are trying to find answers to following questions:
 - Who will use the system?
 - What the system will do?
 - Where and when it will be used?
- In this phase, project team investigates any current systems, identifies improvement opportunities and develops a concept for the new system.
- It has three steps:
 - Analysis strategy
 - Requirements gathering
 - System proposal

Analysis Strategy

- This strategy is developed to guide the project team.
- First, you study the current system and its problems.
- Then you think of ways to design a new-system.
- It is important that you do your analysis correctly, otherwise the new system is just going to be the same and it will not solve any problems.

Requirements Gathering

- This is done by *talking*. You do **interviews**, **workshops** or **questionnaires**.
 - That is why it is very important for a system analyst or a product manager to have good communication skills.
 - You can find additional notes on how to be a better interviewers, etc.
- Here, you also get input from the project sponsor and many other people (stakeholders). Everyone has some ideas and the analyst is the filter.
- All of this is used to develop a concept for the new system.
- All these (analyses, system concept and models) are combined into a document called *system proposal*. This is presented to those who will approve or not.

Requirements Gathering cont.

- The system proposal is the initial document which describes the business requirements that the new system should meet.
- In summary, this part is crucial because you are not building the system for yourself, you are building it for users.
 - However many people may come with unrealistic ideas, be careful.

Interviews

- i. Determine the people to interview
- ii. Establish objectives for the interview
- iii. Develop interview questions
- iv. Prepare for the interview
- v. Conduct the interview
- vi. Document the interview
- vii. Evaluate the interview

Interviews cont.

1. Select the right people and ask the right questions.
2. Determine the areas to be discussed, list the facts you need to gather. Upper management gives you the *big picture* but users can give you specific details.

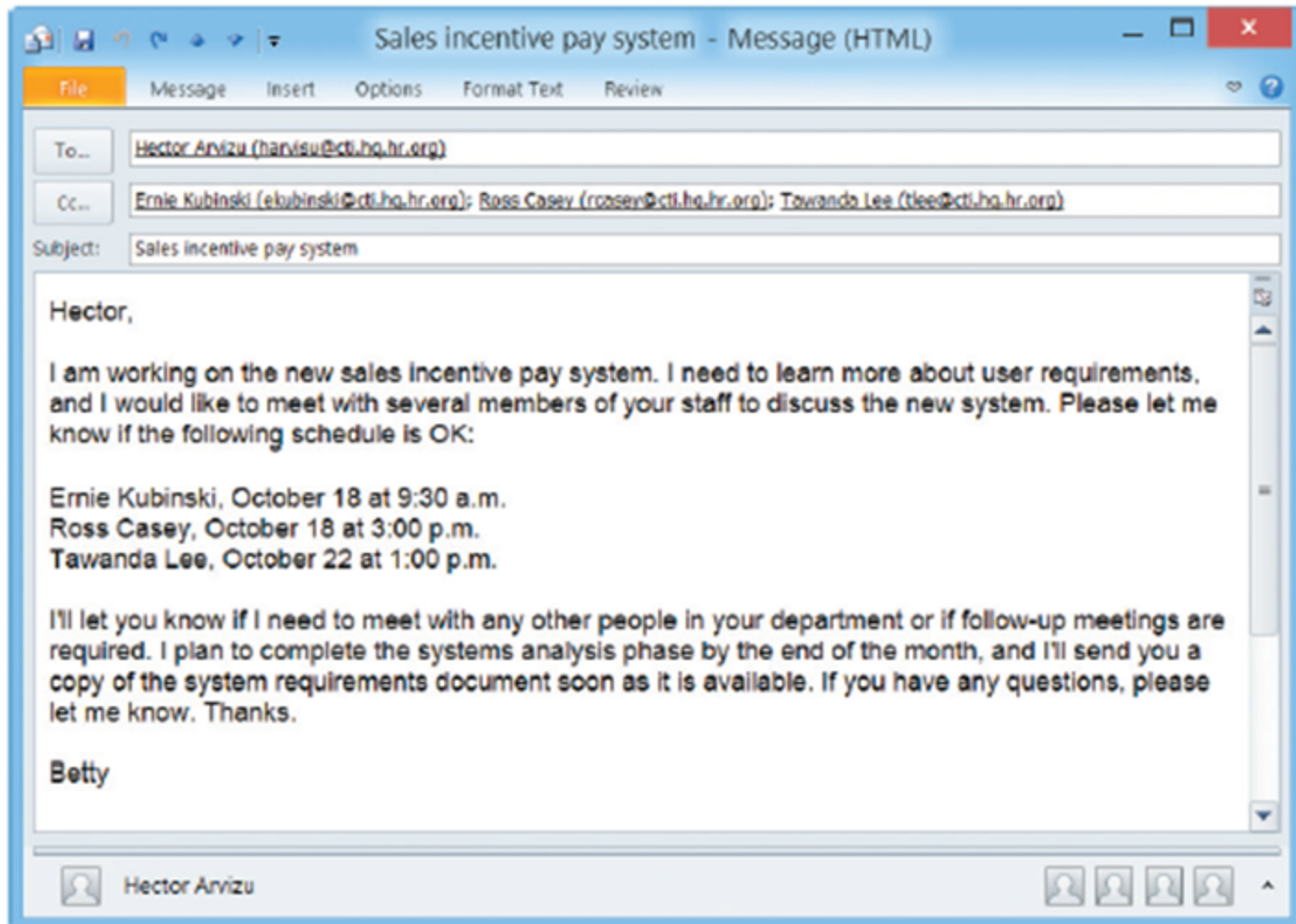
How to develop interview questions

- Decide what to ask and how to phrase it.
- You may choose to ask the same question to different people to *compare* them.
 - Open ended questions encourage spontaneous responses.
 - What are users saying about the new system?
 - How is this task performed?
- Close ended questions can limit the response. These should be used to verify facts.
 - How many personal computers do you have in this department?
 - Do you review the reports before they are sent out?
- Range of response questions use scale - from 1 to 10.
 - How effective was your training?
 - What is the severity of the problem?

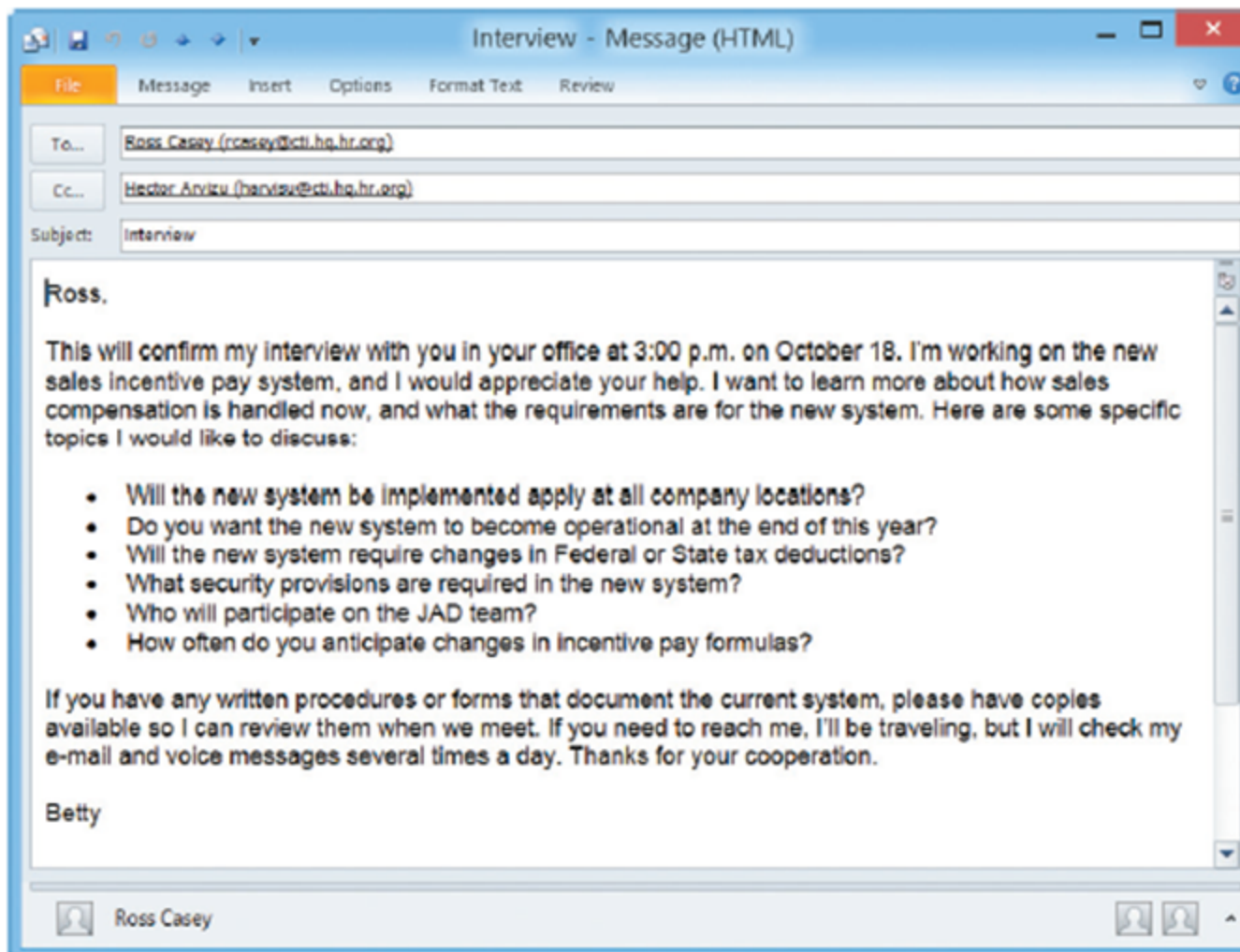
Preparing for the interview

- This is also an important step.
- You must remember that this is not a casual chat, it is a *business meeting*.
- You should set the *time, topic, place* via e-mail or other means (such as Slack).
- If you are not confident, people will think you are slacking and can lose confidence in you or product.

Example



Example



During the interview

- Begin by introducing yourself. Describe the project and explain your interview objectives (Why are you doing it?)
- You need to be listening closely. You should also take fast notes on the answers.
- Give enough time for the other party to think and answer.
- When finished, summarize the session and seek a confirmation to see whether you understood him/her correctly.

After the interview

- Write an after report is possible, especially for yourself.
 - If you are conducting a lot of meetings, this is **very** important. You will forget stuff.
- Send memo to the other party to express your appreciation. Being thankful is good for everyone!
- Note and document everything about it. Also, try to create some action points and take action.

Questionnaires

- There are other *fact finding* techniques such as questionnaires.
- It is usually a fill-in form where people answer questions. However, in software world usually we do interviews.
- The questionnaire must be brief and user-friendly.
 - Provide clear instructions.
 - Arrange questions in logical order. (Simple to more complex)
 - Try not to lead them to specific answers.
 - Limit the use of open-ended questions.
 - etc.

- This is another way of *fact finding*. By the way, keep in mind that it is not that you are going to apply just one of them. You can do them all.
 - Especially this phase, this is done almost every time whether you do interviews or not.
- You can research what your competitors are doing.
 - Maybe they have done a similar thing.

- This phase decides **how** the system will operate in terms of *hardware, software* and *network infrastructure*.
- *User interface, forms, reports* that will be used, *specific programs, databases, files* needed.
 - All of these are decided in this phase.
- Strategic decisions are made in **analysis phase** , here in the **design phase** we determine **how the system will operate**.
- It has four steps.

Design Step 1

- Design strategy.
 - Who is going to program all this?
 - Is it our own programmers, or are we going to outsource them?
 - Are we going to buy an existing software?

Design Step 2

- Architecture design
 - What is the hardware, software and network infrastructure will be used?
 - Are we going to add something new or used the existing infrastructure?
- What is the user interface, are we going to follow a UI system such as Material design, etc.
 - How will users navigate through the system?
 - Menus, on-screen buttons, forms, etc.

Design Step 3

- Which database are we going to use?
 - Are we going to use relational or NoSQL.
 - Maybe we need to use them both?
- Where are we going to store the data? How will we store it?

Design Step 4

- Analyst team develop the program design which defines the programs that need to be written and what will they do.
- So the four steps are:
 - Architecture design, interface design, database & file specs, program design.
- This is the *system specification* that is used by the **programming team** for implementation.
- At the end of design phase, *feasibility analysis* and *project plan* are reexamined and revised and a second decision will be made to decide whether they will continue with the project or terminate it.

Implementation

- Final phase in the SDLC.
- Here, the system is actually **built** (or purchased).
- This is the longest and the most expensive part of most projects.
- It has three steps.
 - Construction, installation and support.

System Construction

- System is built and tested to ensure that it performs as designed.
- Testing is **very important** because *bug fixing* is hard.
 - Most organizations spend more time in testing than writing programs.
 - (This is not actually true.)
 - *There is a popular methodology called TDD. Test Driven Development. Here, you write your own unit tests and it helps a lot but not enough. We will talk about it later.*

Installation

- Here, we turn off the *old* system and turn on the *new* one. There are other terms for this, you can say *deployment* or *merge*.
- There are several approaches here. It depends on your development style.
 - Agile, Waterfall, etc.
- An important thing here is **training**.
 - You have a new system but the users don't know how to use it.
 - Remember, you may build systems for those who are not technologically advanced.
 - So, you need to have a *training plan*.

Support plan

- Analyst team establishes a *support plan* for the system. This usually includes a formal or informal *post-implementation review*.
- Done to identify the major and minor needs for the system if there are any (usually are).