Analysis: Determining System Requirements

System Requirements Determination
Objectives

1. Provide insight into using interviewing to determine system requirements, including the preparation of an interview plan.
2. Show how questionnaires are designed, distributed, and used to determine system requirements.
3. Discuss the advantages and pitfalls of observing workers to determine system requirements.
4. Demonstrate how the analysis of business documents provides system requirements information.
5. Illustrate how Joint Application Design promotes efficient and quick system requirements determination.

The Process of Determining Requirements

• During requirements determination, you and other analysts gather information on what the system should do from as many sources as possible: from developers, administrators and users of the current system, from observing users, from reports, forms and procedures.

• Several characteristics of a good system analyst:
  1. Impertinence *) You should question everything.
  2. Impartiality *) Your role is to find the best solution to a business problem or opportunity.
  3. Relax constrains *) Assume anything is possible and eliminate the infeasible.
  4. Attention to details *) Every fact MUST fit with every other fact.
  5. Reframing *) You MUST challenge your self to look at the organization in different ways.
Traditional Methods for Determining Requirements

Traditional techniques for collecting requirements include

1) interviewing and listening,

2) administering questionnaires

3) observing users, and

4) analyzing procedures and other documents.

1. Interviewing and Listening

- **Interviewing and listening** involves talking with users individually or as a group to discover their views about the current and target systems; it also involves careful preparing an interview outline and guide before conducting the interview.

- **Interviews** are best done when only *a few people are involved*, when you need *open-ended questions* or the questions *vary from individual to individual*, or when a more personal method is needed.
Types of Questions

- **Open-ended questions** are usually used to probe for information for which you cannot anticipate all possible responses or for which you do not know the precise question to ask.

- **Closed-ended questions** provide a range of possible answers from which the interviewee may choose (True/False, Multiple Choice, Ratings a response, Ranking items, etc.)

**Interview Guidelines:**

1) Ask open-ended questions only (!) during interview
2) Do not phrase a question in a way that implies a right or wrong answer.
3) Do not ask questions that imply YES (TRUE) or NO (FALSE) answers.
4) Listen very carefully to what is being said, and take careful notes.
5) Once the interview is over, go back to your office and type up your notes within 24-48 hours.
6) Be careful during the interview not to set expectations about the new or replacement system unless you are sure these features will be part of the delivered system.

Interview Guide is a document for developing, planning and conducting an interview.

Each question in an interview guide can include both verbal and non-verbal information.
Group Interviews

- Interview several key people together

- Advantages
  - More effective use of time
  - Can hear agreements and disagreements at once
  - Opportunity for synergies

- Disadvantages
  - More difficult to schedule than individual interviews

Questionnaires

- Administering questionnaires involves designing a questionnaire and determining who should respond to it; this method is typically used when there are too many key users to interview individually.

- Questionnaires are best when many people are involved, each person is to answer roughly the same questions, and people are remote or do not need personal care.

Interviews versus Questionnaires:

- Interviews provide large amounts of rich, detailed information, but they are expensive to conduct in terms of the time they demand.

- Questionnaires, on the other hand, can reach many people at once, making them relatively less costly than interviews, but the data collected in this way will not be as rich or as plentiful as is the case with interviews.

- Both techniques involve careful planning and execution to be successful.

- Deciding which technique to use will be dependent on such factors as
  - the size and complexity of the information system under study,
  - the size and complexity of the organization in which the system resides,
  - the funding available, and the expertise and preferences of the analysts.
Questionnaire (an example)

Free Web Survey Library
http://www.questionpro.com/a/showLibrary.do?categoryID=6&mode=1
Modern Methods for Collecting System Requirements

- **Joint Application Design (JAD)**
- **CASE Tools**
- **Prototypes**
- **Group Support Systems**

**Joint Application Design**

- Joint Application Design or JAD is a structured process in which users, managers, and analysts work together for several days in a series of intensive meetings to specify or review system requirements.

+ It’s better than traditional techniques because you have all key personnel in one place at one time, saving everyone time and resulting in high levels of system ownership as more people have more of a role in the development process.

- Weaknesses include the level of commitment necessary to make the JAD work, the high degree of required planning, and the typical lack of computer support.

**Joint Application Design (JAD)**

- Intensive group-oriented requirements determination technique
- Team members meet in isolation for an extended period of time
- Highly focused
- Resource intensive
- Started in 1970s by NY Telephone CO. and later by IBM
JAD Session and JAD Participants

<table>
<thead>
<tr>
<th>Role</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Session Leader:</td>
<td>facilitates group</td>
</tr>
<tr>
<td>• Managers:</td>
<td>active, speaking participants</td>
</tr>
<tr>
<td>• Users:</td>
<td>active, speaking participants</td>
</tr>
<tr>
<td>• Systems Analysts:</td>
<td>should mostly listen</td>
</tr>
<tr>
<td>• CIS Staff:</td>
<td>should mostly listen</td>
</tr>
<tr>
<td>• Sponsor:</td>
<td>high-level champion, limited participation</td>
</tr>
<tr>
<td>• Scribe:</td>
<td>record session</td>
</tr>
</tbody>
</table>

Modern Methods for Determining Requirements

- **Prototyping** can be used during requirements determination to collect user requirements and present them in the form of a working system prototype (with less # of functions, GUI main ideas, etc.).
  1) Users can look at, play with, and compare the prototype to their system requirements.
  2) Analysts can then adjust the prototype to better fit what the users have in mind.  
*Ex: Scrum, agile development*

- **CASE tools** can support requirements determination by supporting JAD and prototyping with diagramming, form and report design, repository access, and prototyping tools. The best-suited CASE tools are upper CASE tools.

- **Group support systems** provide unique benefits for group requirements determination through
  1) allowing everyone the *opportunity for equal participation* through typing instead of talking, and
  2) *anonymity* allows the shy and those afraid of criticism to participate.
Prototyping

- Quickly converts requirements to working version of system
- Once the user sees requirements converted to system, will ask for modifications or will generate additional requests
- Most useful when:
  - User requests are not clear
  - Few users are involved in the system
  - Designs are complex and require concrete form
  - History of communication problems between analysts and users
  - Tools are readily available to build prototype

- Drawbacks
  - Tendency to avoid formal documentation
  - Difficult to adapt to more general user audience
  - Sharing data with other systems is often not considered
  - Systems Development Life Cycle (SDLC) checks are often bypassed

Prototyping: Examples

- Concept cars
- Computer games
- CIS
Use-Cases and Use-Case Scenarios

- **USE CASE** is a collection of user scenarios that describe the thread of usage of a system — how different users use an old software system or how they will would like to use a new one.

- Each **SINGLE USECASE SCENARIO** is described from the point-of-view of an “actor”—a person or device that interacts with the software in some way.

- Each **SINGLE** scenario answers the following questions:
  - Who is the primary actor (user), the secondary actor (s)?
  - What are the actor’s goals?
  - What preconditions should exist before the story begins?
  - What main tasks or functions are performed by the actor?
  - What extensions might be considered as the story is described?
  - What variations in the actor’s interaction are possible?
  - What system information will the actor acquire, produce, or change?
  - Will the actor have to inform the system about changes in the external environment?
  - What information does the actor desire from the system?
  - Does the actor wish to be informed about unexpected changes?
It is easy to mix up the definitions of use case and use case scenario.

- A use case (as a class) represents **ALL actions** that are required to enable or abandon a goal. A use case has multiple “paths” that can be taken by any user at any one time.

- A use case scenario is a single path through the use case.

How to create Use-Cases (an algorithm):

1) Identify all USERS of a new system (human beings, non-human beings, other SW applications, online databases, etc.)

2) Identify all FUNCTIONES (FEATURES) of a new system

3) Identify RELATIONS between USERS and FUNCTIONES.

In this course, we will use a lot of:

- **Webster**  current Bradley Online Course Registration System

- **Webster**++ new system that contains proposed-by-you modifications/improvements to be
A Possible Use Case Diagram in for new Webster ++ System

How to create Use-Case (an algorithm):

1) Identify all USERS of a new system (human beings, non-human beings, other SW applications, online databases, etc.)
2) Identify all FUNCTIONES (FEATURES) of a new system
3) Identify RELATIONS between USERS and FUNCTIONES.

Activity Diagram (for one type of users)

Supplements the use case by providing a graphical representation of DETAILED flow of interaction within a specific scenario
Swimlane Diagrams

Allows the modeler to represent the flow of activities described by the use-case and at the same time indicate which actor (if there are multiple actors involved in a specific use-case) or analysis class has responsibility for the action described by an activity rectangle.

Use-Case (a collection of possible cases = paths) and Use-Case Scenario (a single path – given in yellow color) on a SwimLane Diagram.

Source: http://tynerblain.com/blog/2007/04/10/what-are-use-case-scenarios/
Radical Methods: Business Process Reengineering

In some organizations, management is looking for new ways to perform current tasks.

**Business Process Reengineering (BPR)** is the search for, and implementation of, *radical change* in business processes to achieve breakthrough improvements in products and services.

- As part of the BPR effort, **key business processes** should be identified.
- **Key business processes** are the structured, measured set of activities designed to produce a specific output for a particular customer or market.
- **Benefits of BPR** include radical improvements in speed, quality, and customer satisfaction.

Business Process Reengineering: An Example

**Design by Computer:**

One of the most time consuming and expensive business process is the design stage in product development, which had traditionally relied on paper and drafting tools.

Dassault systems has met the challenge of reengineering this process and has created **CATIA CAD system**, the top selling CAD/CAM allows engineers to design and develop products on a computer. This eliminates huge amounts of paper work and slashes the time required to design and develop a new product.

**CATIA CAD System** is used by nearly every air craft manufacturer and was used by Boeing to design the 777.

DaimlerChrysler used **CATIA CAD System** to design the new Jeep Grand Cherokee. By debugging the production line on screen, the company saved months and eliminated $800 million of costs.

Source: http://www.accountingformanagement.com/process_reengineering.htm
Business Re-Engineering due to active implementation of CIS:

Caterpillar Corp.:
in 1994 produced the
same level of output
as did in 1979, but
with 40,000 fewer
employees.

A traditional hierarchical organization with many levels of management

An organization that has been "flattened" by removing layers of management

Business Process Reengineering: An Example
CATERPILLAR WARNS: Bad news is 'converging' and now we have to
make some major changes (Sep 24, 2015)

Business Process Reengineering: An Example

Process reengineering have been used by many companies to deal with a wide variety of problem.

For example, the EMI Records Group was having difficulty filling orders for its most popular CDs. Retailers and recording stars were rebelling—it took the company as much as 20 days to deliver a big order for a hit CD, and then nearly 20% of the order would be missing. Small, incremental improvements would not have been adequate, so the company reengineered its entire distribution process with dramatic effects on on-time delivery and order fill rates.

Reynolds & Reynolds Co. of Dayton, Ohio, produces business forms. Filling an order of a customer used to take 90 separate steps. By reengineering, the number of steps was slashed to 20 and time required to fill an order was cut from three weeks to one week.

Massachusetts General Hospital is even using process reengineering to standardize and improve surgical procedure.

Source: http://www.accountingformanagement.com/process_reengineering.htm

Radical Methods: Disruptive Technologies

**Disruptive technologies** enable the breaking of long-held business rules that inhibit organizations from making radical business changes.

- Disruptive technologies enable companies to apply information technology innovatively.
<table>
<thead>
<tr>
<th>Topic # 7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Determining System Requirements:</strong></td>
</tr>
<tr>
<td><strong>Homework Assignment: 2 exercises</strong></td>
</tr>
<tr>
<td><strong>Required outcome: a single DOC file</strong></td>
</tr>
</tbody>
</table>