Modern Systems Analysis and Design

Fifth Edition

Chapter 3

Analysis

Determining System Requirements

Learning Objectives

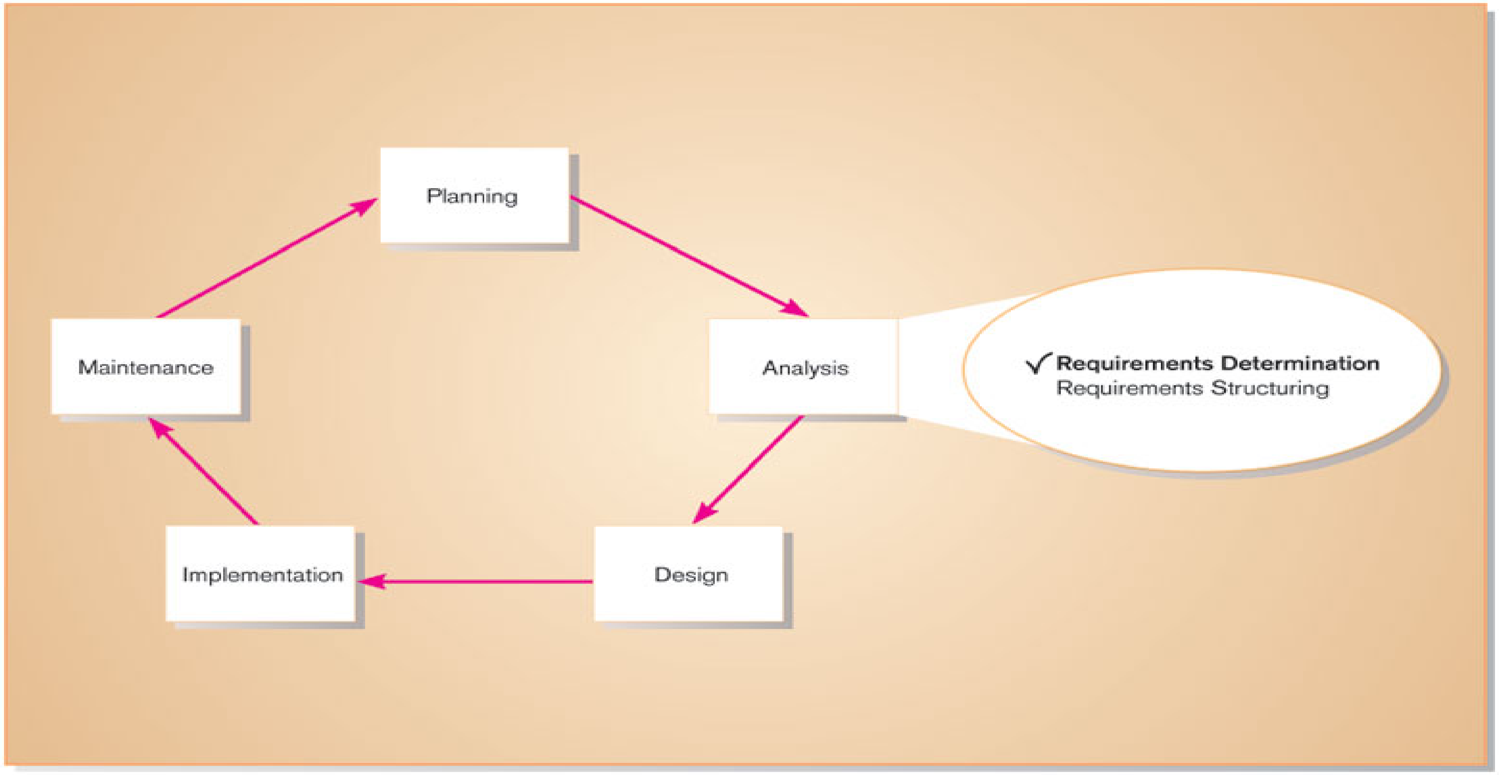
- ✓ Describe options for **designing and conducting interviews** and **develop a plan** for conducting an interview to determine system requirements
- ✓ **Design, distribute, and analyze** questionnaires to determine system requirements
- ✓ Explain **advantages** and pitfalls of observing workers and analyzing business documents to determine requirements

Learning Objectives

- ✓ Explain how computing can provide support for requirements determination
- ✓ Learn about Joint Application Design (JAD)
- ✓ Use prototyping during requirements determination
- ✓ Select the appropriate methods to elicit system requirements

Performing Requirements Determination

Figure 6-1 Systems development life cycle with analysis phase highlighted



Performing Requirements Determination

- Gather information on what system should do from many sources
  - Users
  - Reports
  - Forms
  - Procedures

Performing Requirements Determination

- **Characteristics for gathering requirements**
  - **Impertinence**
    - **Question everything**
  - **Impartiality**
    - **Find the best organizational solution**
  - **Relaxation of constraints, assume anything is possible**
  - **Attention to detail, every fact must fit with every other fact**
  - **Reframing**
    - **View the organization in new ways,**

Deliverables and Outcomes

- **Types of deliverables:**
  - From interviews and observations - interview transcript observation notes, meeting minutes
  - From existing written documents - mission and strategy statements, business forms, procedure manuals, job descriptions, training manuals, system documentation, flowcharts
  - From computerized sources – Joint Application Design session results, CASE repositories, reports from existing systems, displays and reports from system prototype.

  - **Information collected from users**
  - **Existing documents and files**
  - **Computer-based information**
  - **Understanding of organizational components**
    - **Business objective**
    - **Information needs**
    - **Rules of data processing**
    - **Key events**

Traditional Methods for Determining Requirements

- **Interviewing and Listening**
  - **Gather facts, opinions and speculations**
  - **Observe body language and emotions**
  - **Guidelines**
    - **Plan**
      - **Checklist**
      - **Appointment**
    - **Be neutral**
    - **Listen**
    - **Seek a diverse view**

Traditional Methods for Determining Requirements

- Interviewing (Continued)
  - Interview Questions
    - Open-Ended
      - No pre-specified answers
    - Close-Ended
      - Respondent is asked to choose from a set of specified responses
  - Additional Guidelines
    - Do not phrase questions in ways that imply a wrong or right answer
    - Listen very carefully to what is being said
    - Type up notes within 48 hours
    - Do not set expectations about the new system

Traditional Methods for Determining Requirements

Administering Questionnaires

- More cost-effective than interviews
- Choosing respondents
  - Should be representative of all users
  - Types of samples
    - Convenient, local site.
    - Random sample
    - Purposeful sample, people who satisfy certain criteria.
    - Stratified sample, random set of people from many hierarchical levels.

Traditional Methods for Determining Requirements

- **Questionnaires**
  - **Design**
    - **Mostly closed-ended questions**
    - **Can be administered over the phone or in person**
  - **Vs. Interviews**
    - **Interviews cost more but yield more information**
    - **Questionnaires are more cost-effective**
    - **See table 7-4 for a complete comparison**

Traditional Methods for Determining Requirements

- **Interviewing Groups**
  - **Advantages**
    - **More effective use of time**
    - **Enables people to hear opinions of others and to agree or disagree**
  - **Disadvantages**
    - **Difficulty in scheduling**

- **Nominal Group Technique (NGT)** A facilitated process that supports idea generation by groups**.**
  - Process
    - Members come together as a group, but initially work separately.
    - Each person writes ideas.
    - Facilitator reads ideas out loud, and they are written on a blackboard or flipchart.
    - Group openly discusses the ideas for clarification.
    - Ideas are prioritized, combined, selected, reduced.

6.12

Traditional Methods for Determining Requirements

- **Directly Observing Users**
  - Watching users do their jobs
  - Obtaining more firsthand and objective measures of employee interaction with information systems.
  - Can cause people to change their normal operating behavior.
  - Time-consuming and limited time to observe.

Analyzing Procedures and Other Documents

- **Types of information to be discovered when analyzing a document:**
  - **Problems with existing system**
  - **Opportunity to meet new need**
  - **Organizational direction**
  - **Names of key individuals**
  - **Values of organization**
  - **Special information processing circumstances**
  - **Reasons for current system design**
  - **Rules for processing data**

Analyzing Procedures and Other Documents

- **Four types of useful documents to SA:**
  - **Written work procedures**
    - For an individual or work group.
    - Describes how a particular job or task is performed.
    - Includes data and information used and created in the process
  - **Business form**
    - **Explicitly indicate data flow in or out of a system**
  - **Report generated by current systems**
    - **Enables the analyst to work backwards from the report to the data that generated it**
  - **Description of current information system, how they were designed and how they work**

Forma and Informal system

- **Formal Systems**: the official way a system works as described in organizational documentation (i.e. work procedure).
- **Informal Systems**: the way a system actually works (i.e. interviews, observations).

Modern Methods for Determining Requirements

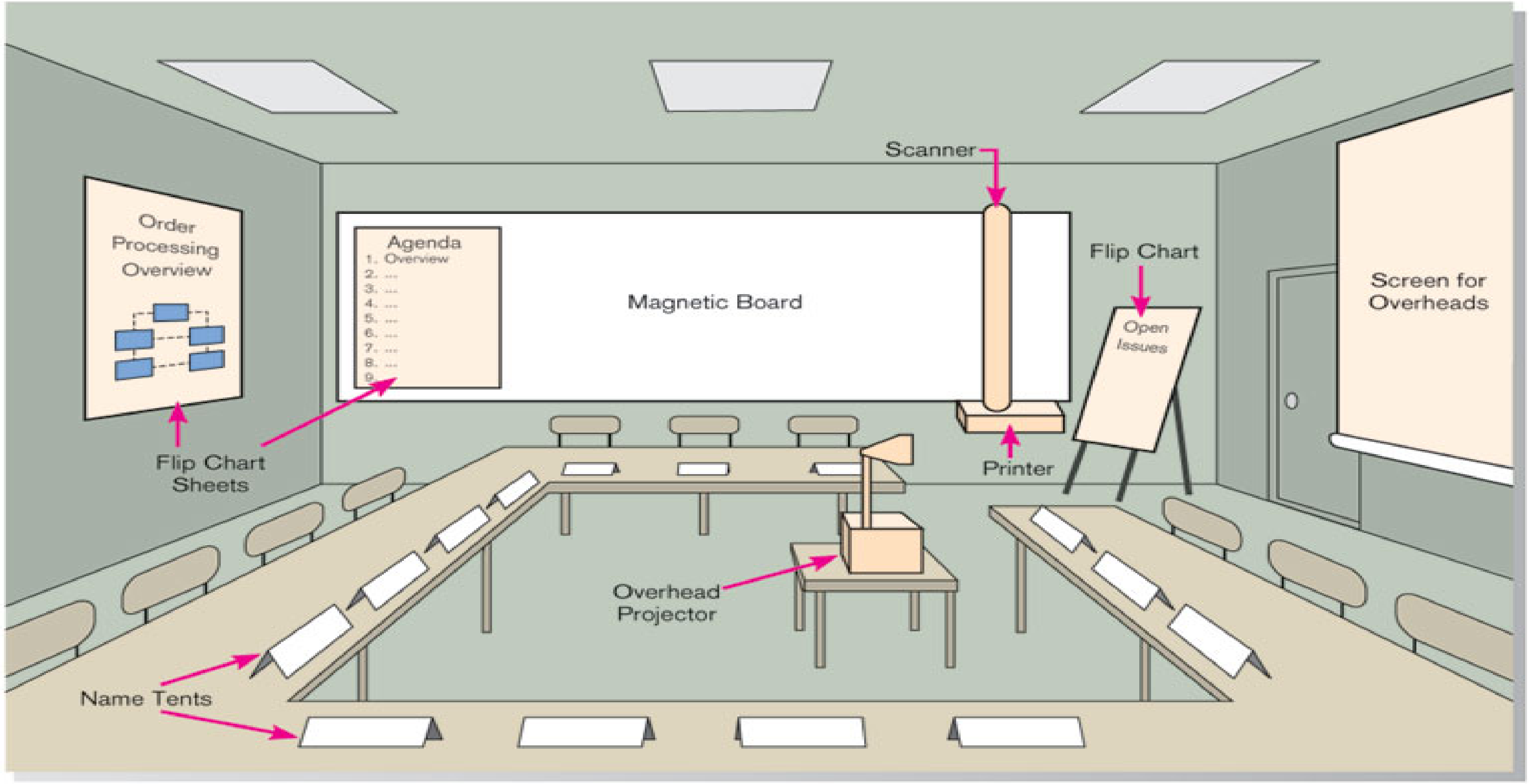
- **Joint Application Design (JAD)**
  - Brings together key users, managers and systems analysts
  - Purpose: collect system requirements simultaneously from key people
  - Conducted off-site
- **Prototyping**
  - Repetitive process
  - Basic version of system is built
  - Refine understanding of system requirements in concrete terms.
  - Goal: to develop concrete specifications for ultimate system

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 by IBM in 1970s.

JAD

Figure 6-6 Illustration of the typical room layout for a JAD



Source: Adapted from Wood and Silver, 1995.

JAD

- JAD Participants:
  - **Session Leader**: facilitates group process.
  - **Users:** active, speaking participants
  - **Managers**: active, speaking participants
  - **Sponsor**: high-level champion, limited participation.
  - **Systems Analysts**: should mostly listen.
  - **Scribe**: record session activities.
  - **IS Staff**: should mostly listen.
- End Result
  - Documentation detailing existing system.
  - Features of proposed system.

Joint Application Design (JAD)

- **CASE Tools During JAD**
  - **Upper CASE tools are used**
  - **Enables analysts to enter system models directly into CASE during the JAD session**
  - **Screen designs and prototyping can be done during JAD and shown to users**

Joint Application Design (JAD)

- Supporting JAD with GSS
  - Group support systems (GSS) can be used to enable more participation by group members in JAD
  - Facilitate sharing of ideas and voicing of opinions about system requirements.
  - Members type their answers into the computer
  - All members of the group see what other members have been typing

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**

Prototyping

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

6.24

Business Process Reengineering (BPR)

- **Search for and implementation of radical change in business processes to achieve breakthrough improvements in products and services**
- **Goals**
  - Reorganize complete flow of data in major sections of an organization.
  - Eliminate unnecessary steps.
  - Become more responsive to future change.
  - Combine steps

Business Process Reengineering (BPR)

- **Identification of processes to Reengineer**
  - **Key business processes**
    - **Set of activities designed to produce specific output for a particular customer or market**
    - **Focused on customers and outcome**
    - **Key business process includes all activities of design, build, deliver and support a product.**

Business Process Reengineering (BPR)

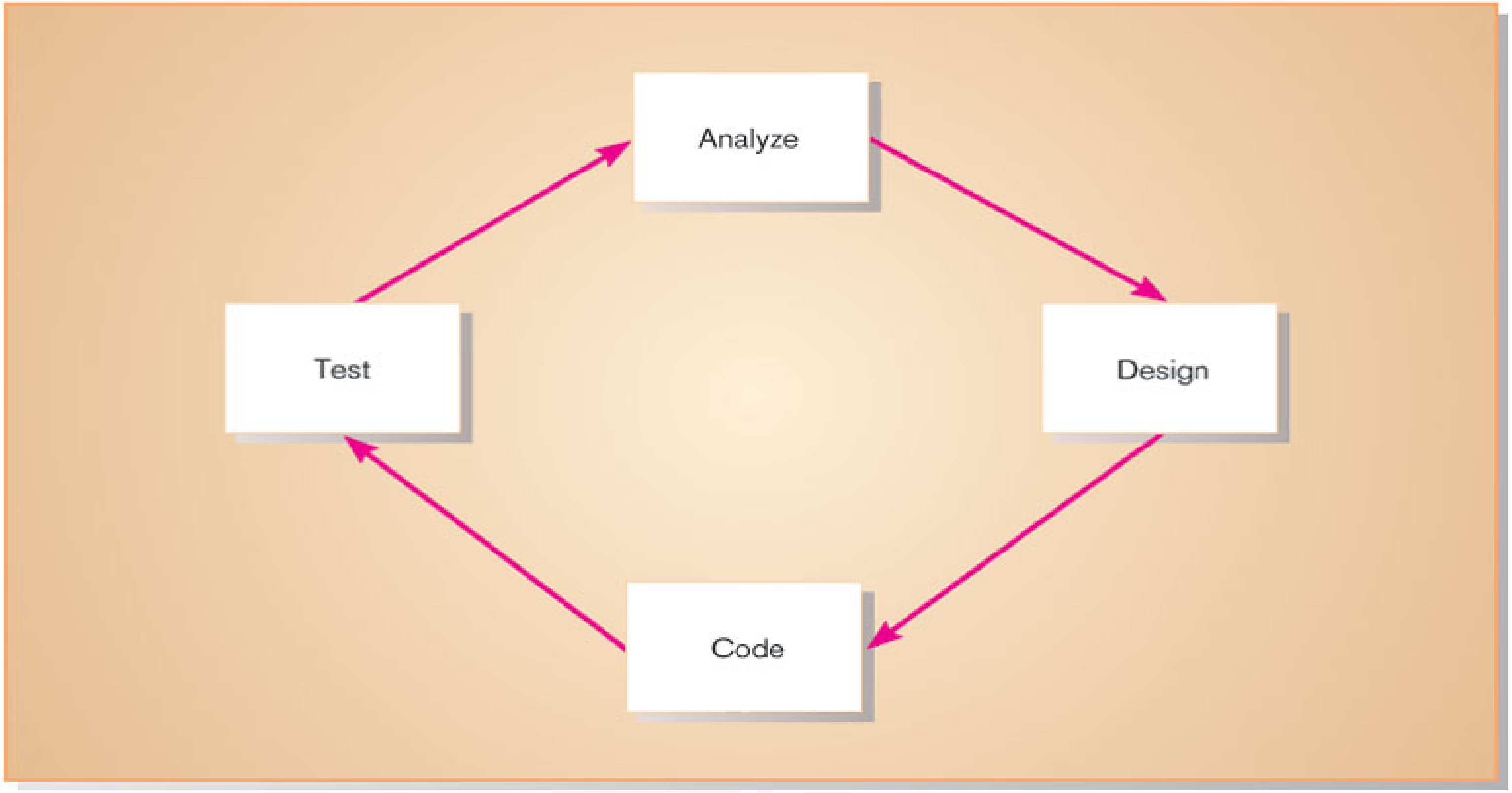
- Identify specific activities that can be improved through BPR, once it have been identified, Information Technology must be applied to radically improve business process.
- **Disruptive technologies**
  - are technologies that enable the breaking of long-held business rules that inhibit organizations from making radical business changes. (**decision support tools, wireless data communication, high performance computing can provide real-time updating**)
  - **See table 6-6. More Examples.**

Requirements determining using Agile Methodologies

- **Continual user involvement**
  - Replace traditional SDLC waterfall with iterative analyze – design – code – test cycle
- **Agile usage-centered design**
  - Focuses on user goals, roles, and tasks
  - Gather a group of people all stakeholders in one room.
  - Give everyone a chance to talk about current and new system.
  - Determine user roles and goals
  - Determine task needs to be completed to achieve the goal.
  - Task cards will be grouped together based on similarity.
  - For each task, list steps that are necessary to complete the step.
  - Treat each set of tasks to be supported by a single aspect of user interface (partition task)
  - Prototype and refine the prototype
- **The Planning Game**
  - Based on eXtreme programming
  - Exploration, steering, commitment

Continual User Involvement

Figure 6-7 The iterative analysis–design–code–test cycle

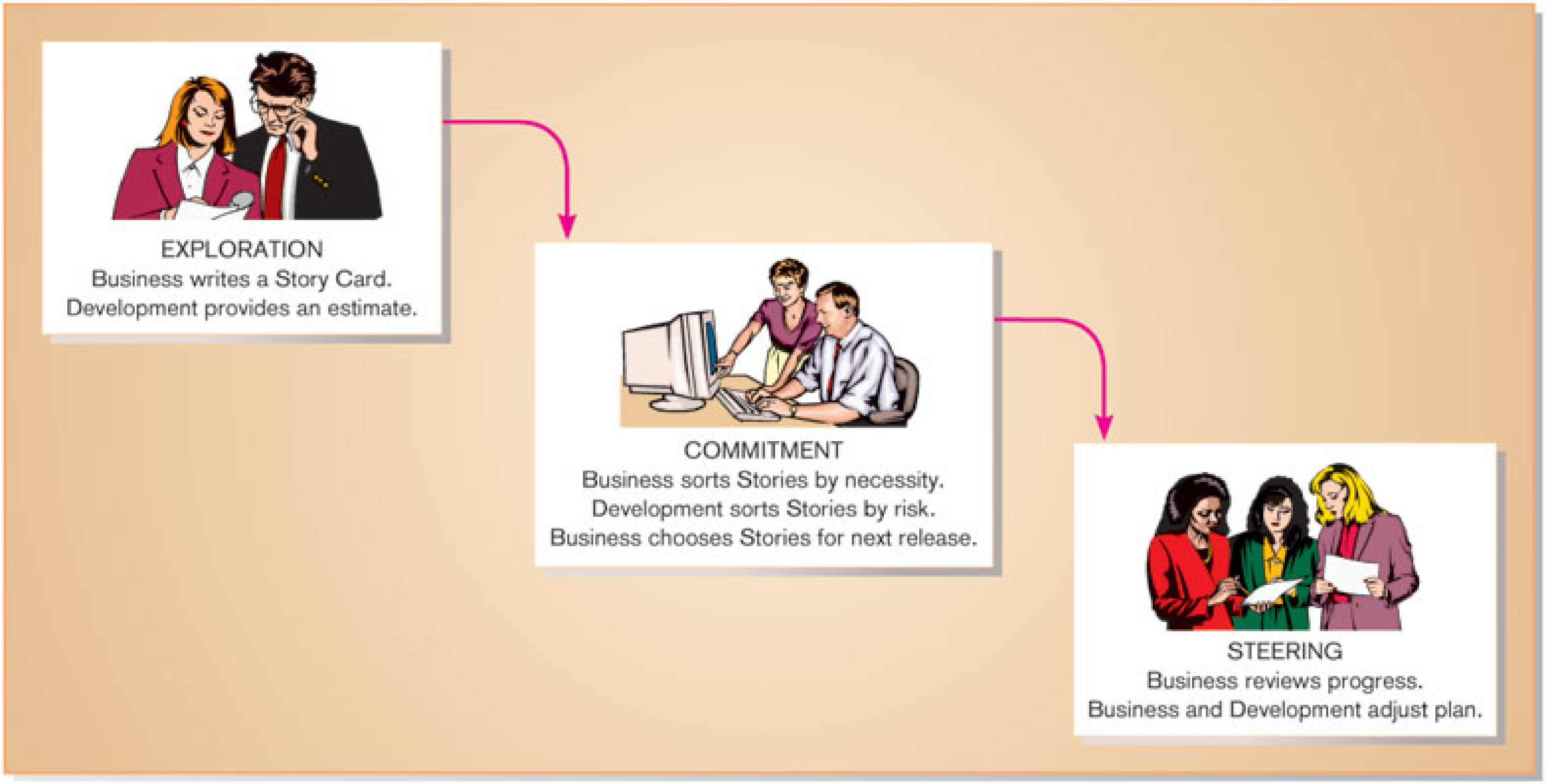


Agile Usage-Centered Design Steps

- Gather group of programmers, analysts, users, testers, facilitator.
- Document complaints of current system.
- Determine important user roles.
- Determine, prioritize, and describe tasks for each user role.
- Group similar tasks into interaction contexts.
- Associate each interaction context with a user interface for the system, and prototype the interaction context.
- Step through and modify the prototype.

The Planning Game from eXtreme Programming

Figure 6-8 eXtreme Programming's Planning Game



EXPLORATION
Business writes a Story Card.
Development provides an estimate.

COMMITMENT
Business sorts Stories by necessity.
Development sorts Stories by risk.
Business chooses Stories for next release.

STEERING
Business reviews progress.
Business and Development adjust plan.

Requirements determining using Agile Methodologies

- The planning game from extreme programming: (phases)
  - Exploration, business create story cards, development with an estimation of how long it would take to implement.
  - Commitment, sort the story cards and split them to essential, not essential, and nice to have.
  - Steering, to see how the development process is progressing.

The planning game is followed by iteration planning game, played by programmers.