Software Engineering
Requirements

Dr. Young-Woo Kwon

Overview: Requirements Analysis

- What is *requirement*?
- Classification of requirements
- How to gather requirements?

Requirements

- Definition
  - The features that your application must provide
- Focusing on the ***WHAT***, not on the ***HOW***

Requirements Analysis is Hard

- Major causes of project failures
  - Incomplete requirements
  - Changing requirements
  - Poor user input
- Essential solutions
  - Classification of requirements
  - Requirements analysis
    - Gathering, refining, and reordering requirements

Requirements Analysis Process

- Gathering Requirements
  - Interviews
  - Joint Application Development (JAD)
  - Questionnaires
  - Document analysis
  - Observation
- Refining Requirements
  - Copy existing systems
  - Brainstorm

Requirements Analysis Process

- Reordering Requirements
  - UML
  - User Stories
  - Use Cases
  - Prototypes
  - Requirements Specification
- Validation and Verification
  - Validation: Are we doing the right things?
  - Verification: Are we doing the things right?
- Changing Requirements

REQUIREMENTS DETERMINATION

Introduction

- Requirements determination
  - The single most critical step of the entire SDLC
  - Changes can be made easily in this stage
  - Most (>50%) system failures are due to problems with requirements
  - The iterative process is effective because:
    - Small batches of requirements can be identified and implemented incrementally
    - The system will evolve over time

Requirements abstraction

"If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not pre-defined. **The requirements must be written** so that several contractors can **bid for the contract**, offering, perhaps, different ways of meeting the client organization's needs. Once a contract has been awarded, **the contractor must write a system definition** for the client in more detail so that the client understands and can validate what the software will do**. Both of these documents may be called the requirements document** for the system."

소프트웨어 발주 프로세스

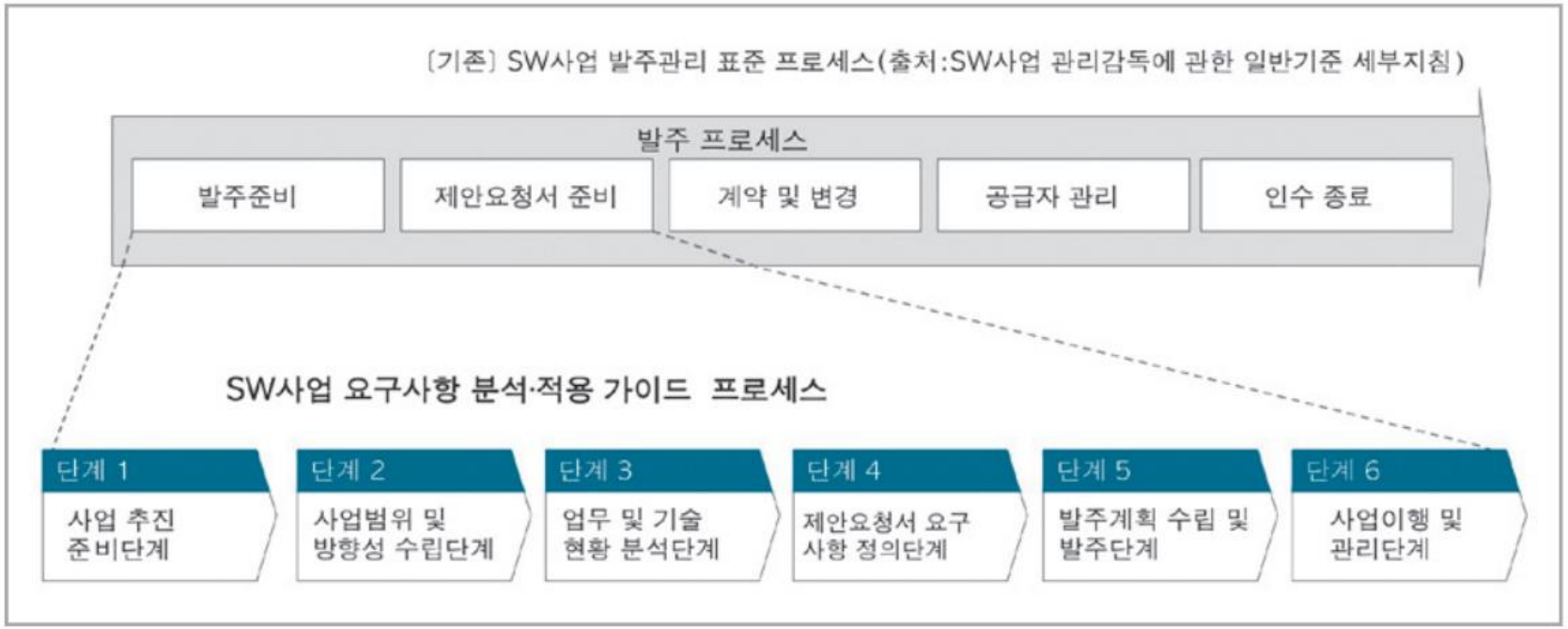


그림 Ⅰ-2 소프트웨어사업 요구사항 분석·적용 가이드개념도

Types of requirement

- User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints.
- System requirements
  - A structured document setting out detailed descriptions of the system's functions, services and operational constraints.
  - Defines what should be implemented so may be part of a contract between client and contractor.

Requirements Determination

- Purpose: to convert high level business requirements (from the system request) into detailed requirements that can be used as inputs for creating models
- What is a requirement?
  - A statement of what the system must do or a characteristic it must have
  - Will later evolve into a technical description of how the system will be implemented
- Types:
  - Functional: relates to a process or data
  - Non-functional: relates to performance or usability

Requirements Definition

- Functional & non-functional requirements listed in outline format
- May be prioritized
- Provides information needed in subsequent workflows
- Defines the scope of the system

Sample of Requirements Definition

Nonfunctional Requirements

1. Operational Requirements
1.1. The system will operate in Windows environment.
1.2. The system should be able to connect to printers wirelessly.
1.3. The system should automatically back up at the end of each day.

2. Performance Requirements
2.1. The system will store a new appointment in 2 seconds or less.
2.2. The system will retrieve the daily appointment schedule in 2 seconds or less.

3. Security Requirements
3.1. Only doctors can set their availability.
3.2. Only a manager can produce a schedule.

4. Cultural and Political Requirements
4.1. No special cultural and political requirements are anticipated.

Functional Requirements

1. Manage Appointments
1.1. Patient makes new appointment.
1.2. Patient changes appointment.
1.3. Patient cancels appointment.

2. Produce Schedule
2.1. Office Manager checks daily schedule.
2.2. Office Manager prints daily schedule.

3. Record Doctor Availability
3.1. Doctor updates schedule

Classification of Requirements

- **Functional**: features, capabilities, security
  - E.g., the system reads employee records and prints paychecks
  - All other requirements are non-functional
- **Usability**: human factors, help, documentation
  - E.g., text on the display must be visible from 1 meter
- **Reliability**: frequency of failure, recoverability, predictability
  - E.g., when doing search, the radar should have 28 hours mean time between failures

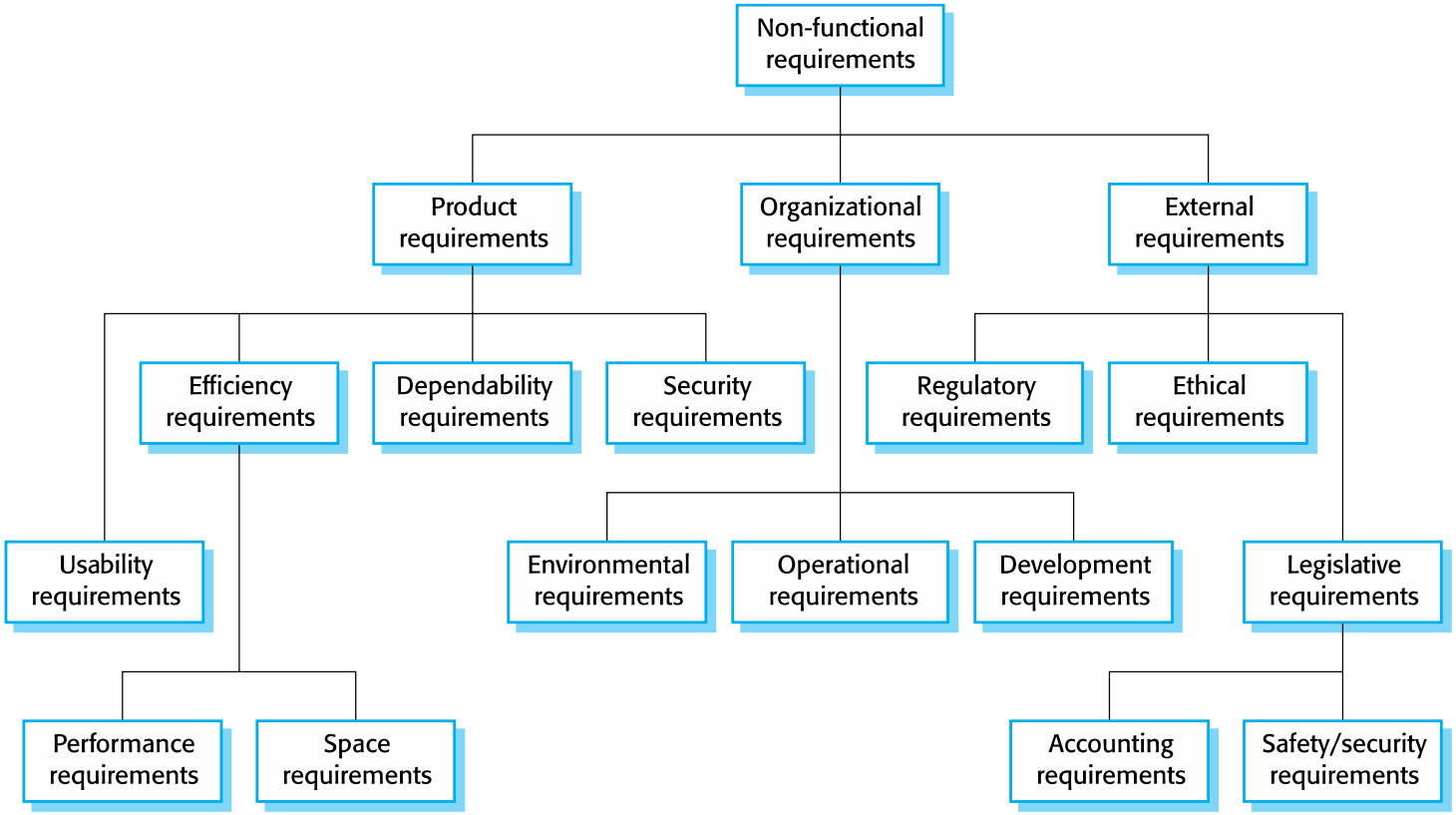
Classification of Requirements

- **Performance**: response times, throughput, accuracy, availability, resource usage
  - E.g., the server response time is < 1 sec for 90% of the accesses
- **Supportability**: adaptability, maintainability, internationalization, configurability
  - The system should allow frequent and easy changes in the network configuration

Extended Requirements

- **Design requirements**
  - The system should mirror database in case the main DB crashes
- **Implementation requirements**
  - Must use Java on Linux
- **Interface requirements**
  - Format of the data that will be exchanged
- **Physical requirement**
  - The system requires a minimum amount of processing power, easy portability, etc.

Types of nonfunctional requirement



Non-functional requirements implementation

- Non-functional requirements may affect the overall architecture of a system rather than the individual components.
  - For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.
- A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define system services that are required.
  - It may also generate requirements that restrict existing requirements.

Non-functional classifications

- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- Organisational requirements
  - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.
- External requirements
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements

Examples of nonfunctional requirements

Product requirement
The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

Organizational requirement
Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.

External requirement
The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

표 Ⅰ-1 **소프트웨어사업 유형별 제안요청서 요구사항 분류 항목**

제안요청서 작성 요구사항	내 용	사업유형			
		소프트웨어 개발	시스템 운용환경	유지관리	정보화 전략계획
시스템 장비구성 요구사항	– 목표사업수행을 위해 필요한 하드웨어, 소프트웨어, 네트워크 등의 도입 장비 내역 등 시스템 장비 구성에 대한 요구사항을 기술함	●	●		
기능 요구사항	– 목표 시스템(사업)이 반드시 수행해야 하거나 목표 시스템을 이용하여 사용자가 반드시 수행할 수 있어야 하는 기능(동작)에 대하여 기술함 – 단, 개별 기능요구사항은 전체 시스템의 계층적 구조분석을 통해 단위 업무별 기능구조를 도출한 후, 이에 대한 세부 기능별 상세 요구사항을 작성하는 것을 원칙으로 하며, 기능 수행을 위한 데이터 요구사항과 연계를 고려하여 기술함	●	◐		◐
성능 요구사항	– 목표 시스템의 처리속도 및 시간, 처리량, 동적 · 정적용량, 가용성 등 성능에 대한 요구사항을 기술함	●	●		
인터페이스 요구사항	– 목표시스템과 외부를 연결하는 시스템 인터페이스와 사용자 인터페이스에 대한 요구사항을 기술한 것으로 타 소프트웨어, 하드웨어 및 통신 인터페이스, 타 시스템들과의 정보교환에 이용되는 프로토콜과의 연계도 포함하여 기술함 – 단, 인터페이스 요구사항의 경우 사용자 편의성, 사용자 경험 등의 사용자 중심의 요구사항을 기술함	●	◐		

제안요청서 작성 요구사항	내 용	사업유형			
		소프트웨어 개발	시스템 운용환경	유지관리	정보화 전략계획
데이터 요구사항	– 목표 시스템의 서비스에 필요한 초기자료 구축 및 데이터 변환을 위한 대상, 방법, 보안이 필요한 데이터 등 데이터를 구축하기 위해 필요한 요구사항을 기술함	●	●		
테스트 요구사항	– 도입되는 장비의 성능 테스트(BMT) 또는 구축된 시스템이 계획된 목표 대비 제대로 운영되는가를 테스트하고, 점검하기 위한 테스트 요구사항을 기술함	●	●		
보안 요구사항	– 정보 자산의 기밀성과 무결성을 확보하기 위해 목표 시스템의 데이터 및 기능, 운영 접근을 통제하기 위한 요구사항을 기술함	●	●	●	
품질 요구사항	– 목표 사업의 원활한 수행 및 운영을 위해 관리가 필요한 품질 항목, 품질 평가 대상 및 목표에 대한 요구사항을 기술함 – 신뢰성, 사용성, 유지관리성, 이식성, 보안성으로 구분하여 기술함	●	●	●	
제약사항	– 목표시스템 설계, 구축, 운영과 관련하여 사전에 파악된 기술 · 표준 · 업무 · 법제도 등 제약조건 등을 파악하여 기술함	●	●	●	●
프로젝트 관리 요구사항	– 프로젝트의 원활한 수행을 위한 관리 방법 및 추진 단계별 수행방안에 대한 요구사항을 기술함	●	●	●	●
프로젝트 지원 요구사항	– 프로젝트의 원활한 수행을 위해 필요한 지원사항 및 방안에 대한 요구사항을 기술함 – 시스템/서비스 안정화 및 운영, 교육훈련 및 기술지원, 하자보수 또는 유지관리 요구사항 등을 기술함	●	●	●	●

요구사항 예시

표 Ⅱ-6 **소프트웨어개발사업의 기능 요구사항 작성 예시**

요구사항 고유번호		SFR-FA-001		
요구사항 명칭		발의내역 관리		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	– 회계관리 시스템 – 발의관리 – 발의내역관리 – 일반발의, 가지급금, 기타발의(회의비 신청서 등) 내역을 관리하고, 발의내역에 대한 전자결재시스템과 연계하는 기능		
	세부 내용	① 사용 목적, 사업계정, 계정, 소요예산 등을 입력하는 기능 ② 계약 업무, 소액구매, 전문가 수당, 법인카드, 출장 등의 발의내역 등록 기능 ③ 급여/퇴직금 산정, 국내/해외 출장비 신청, 평가비 산정, 구매 요청(소액, 단가)시 자동 발의 기능 ④ 발의내역 등록 후, 품의서 생성 기능을 연계하여 발의내역을 품의서 내용에 반영하는 기능 – 연계기능: 급여시스템, 인사시스템, 평가시스템, 계약시스템 – 산출물: 발의내역서		

요구사항 고유번호		PER-001		
요구사항 명칭		동시 사용자 접속 수		
요구사항 분류		성능	응락수준	필수
요구사항 상세설명	정의	– 전자결재 시스템		
	세부 내용	– 시스템 당 동시 사용자 500명 이상 지원해야 하고 성능이 저하되지 않아야 함 – 지난 5분 이상 요청한 사람들만 로그인 사용자로 간주함		

요구사항 고유번호		PER-002		
요구사항 명칭		웹 페이지 디스플레이 시간		
요구사항 분류		성능	응락수준	필수
요구사항 상세설명	정의	– 목표 시스템 웹 페이지		
	세부 내용	– 정보 요청에서 결과가 조회되는 것에 대한 응답시간을 의미함 – 시스템이 만드는 각 웹 페이지의 경우, 사용자가 요청한 시각으로부터 3초 내에 완전히 디스플레이 되어야 함 – 성능 예외사항: 이 요구사항은 임의의 선택 기준이 허용되는 대량의 데이터에 대한 질의 및 플래시가 있을 경우에는 적용되지 않으며, 한 개 이상의 큰 이미지 (이미지 500KB 이상) 및 동영상을 가지고 있는 페이지에는 적용되지 않음. 또한 시스템을 사용하는 사용자 수가 동시 사용자 용량의 90%를 초과하는 경우에는 적용되지 않음		

요구사항 고유번호		SIR001		
요구사항 명칭		CRM과 VOC 연계		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세설명	정의	VOC 시스템(송신)과 CRM 시스템(수신)간의 인터페이스		
	세부 내용	– VOC 시스템을 통해 들어온 고객 불만 관련 정보를 CRM 시스템에 일 1회 전달함 – 1일 예상 건수는 500회 정도임		
기타 고려 사항		– (프로젝트 범위 외)VOC 시스템 담당자는 VOC 시스템에서 CRM 시스템으로 고객 불만 관련 정보를 보내도록 프로그램을 수정함 – 목표 시스템 구축 업체는 CRM 시스템을 수정하여 해당 정보를 수정하며, VOC 시스템과의 연동 테스트를 함께 수행해야 함		
산출정보		– CRM 시스템에 전달된 고객 불만 정보		
요구사항 출처		– OOO 책임		

요구사항 고유번호		SIR003		
요구사항 명칭		온라인도움말		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세설명	정의	온라인 도움말 제공		
	세부 내용	– 사용자 기능은 온라인 도움말을 제공해야 함 – 사용자가 별도의 교육을 받지 않더라도 온라인 도움말을 이용하여 사용자 기능을 이용할 수 있어야 함 – 사용자 기능 사용시 오류가 발생하면 에러 메시지 기능이 제공되어야 함		
주석		– 사용성은 기능 및 품질요구사항과의 연관성이 높음. 따라서 이들 간에는 추적관리가 이루어져야 함		
요구사항 출처		– OOO 수석		

Determining Requirements

- Business & IT personnel need to collaborate
- Strategies for requirements determination:
  - Problem analysis
  - Root cause analysis
  - Duration analysis
  - Activity-based costing
  - Informal benchmarking
  - Outcome analysis
  - Technology analysis
  - Activity elimination

Requirements Analysis Strategies

- Problem analysis
  - Ask users to identify problems with the current system
  - Ask users how they would solve these problems
  - Good for improving efficiency or ease-of-use
- Root cause analysis
  - Focus is on the cause of a problem, not its solution
  - Create a prioritized list of problems
  - Try to determine their causes
  - Once the causes are known, solutions can be developed

Requirements Analysis Strategies(Cont.)

- Duration analysis
  - Determine the time required to complete each step in a business process
  - Compare this to the total time required for the entire process
  - Large differences suggest problems that might be solved by:
    - Integrating some steps together
    - Performing some steps simultaneously (in parallel)
- Activity-based costing
  - Same as duration analysis but applied to costs
- Informal benchmarking
  - Analyzes similar processes in other successful organizations

Requirements Analysis Strategies(Cont.)

- Outcome analysis
  - What does the customer want in the end?
- Technology analysis
  - Apply new technologies to business processes & identify benefits
- Activity elimination
  - Eliminate each activity in a business process in a “force-fit” exercise

Requirements Gathering Techniques

- Process is used to:
  - Uncover all requirements (those uncovered late in the process are more difficult to incorporate)
  - Build support and trust among users
- Which technique(s) to use?
  - Interviews
  - Joint Application Development (JAD)
  - Questionnaires
  - Document analysis
  - Observation

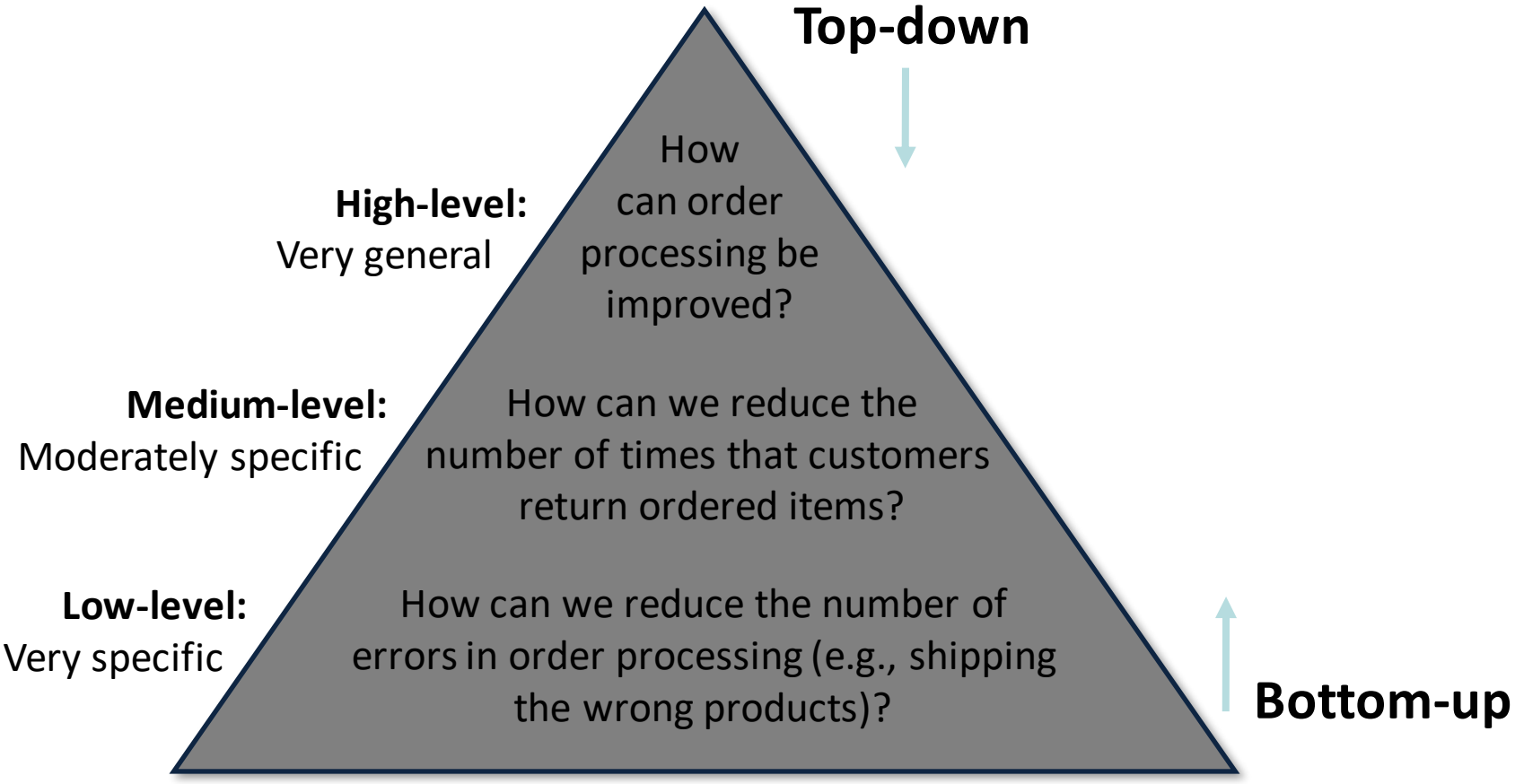
Interviews

- Most popular technique—if you need to know something, just ask
- Process:
  - Select people to interview & create a schedule
  - Design interview questions (Open-ended, closed-ended, & probing types of questions)
  - Prepare for the interview (Unstructured vs. structured interview organized in a logical order)
  - Conduct the interview (Top-down vs. bottom-up)
  - Follow-up after the interview

Question Types

Types of Questions	Examples
Closed-ended questions	• How many telephone orders are received per day? • How do customers place orders? • What information is missing from the monthly sales report?
Open-ended questions	• What do you think about the current system? • What are some of the problems you face on a daily basis? • What are some of the improvements you would like to see in a new system?
Probing questions	• Why? • Can you give me an example? • Can you explain that in a bit more detail?

Interviewing Strategies



Post-Interview

- Prepare notes and send to the interviewee for verification

Interview Notes Approved by: Linda Estey

Person Interviewed: Linda Estey,
Director, Human Resources

Interviewer: Barbara Wixom

Purpose of Interview:

- Understand reports produced for Human Resources by the current system
- Determine information requirements for future system

Summary of Interview:

- Sample reports of all current HR reports are attached to this report. The information that is not used and missing information are noted on the reports.
- Two biggest problems with the current system are:
  1. The data are too old (the HR Department needs information within two days of month end; currently information is provided to them after a three-week delay)
  2. The data are of poor quality (often reports must be reconciled with departmental HR database)
- The most common data errors found in the current system include incorrect job level information and missing salary information.

Open Items:

- Get current employee roster report from Mary Skudrna (extension 4355).
- Verify calculations used to determine vacation time with Mary Skudrna.
- Schedule interview with Jim Wack (extension 2337) regarding the reasons for data quality problems.

Detailed Notes: See attached transcript.

인터뷰 예시

운영자 인터뷰 질의서

부서		장소	
진행자		일시	
참석자			

구분	질문	답변
응용시스템 운영/관리 부분	1) 현재 담당하고 계신 업무를 설명해 주시기 바랍니다.	
	2) 담당하고 계신 업무(응용시스템)의 주된 사용자(대상자)는 어느 부서인가요?	
	3) 담당하고 계신 현 어플리케이션의 문제점 또는 이슈에는 어떤 것이 있습니까? 예) - 프로그램 관점: 공통모듈의 활용도가 낮음.(에러메시지 처리 등) - 데이터 관점: 동일 데이터 중복 저장 및 데이터 클린징 요건 발생 - 기술요소 관점: 네트워크 노후화로 인한 거래 처리 시간 지연 - 운영 관점: 업무 요건 변경 시 프로그램 및 데이터의 영향도 파악이 어려움	
	4) 어플리케이션의 기능이 유사하거나 중복되는 부분이 있으시면 기술하여 주시기 바랍니다.	
	5) 현재 담당하는 업무에서 어플리케이션의 통합 또는 분산이 필요하다고 느끼시는 시스템은 무엇입니까?(현재 중복기능 존재여부 등)	
	6) 응용시스템간의 연관성이 명확히 정의되어 있습니까?(시스템이 수정될 경우, 타 시스템에 어떠한 영향을 미치는지 아는 방법은 무엇입니까?)	
	7) 어플리케이션 운영 시 보안솔루션 및 보안 시스템 운용환경에 대한 전	

그림 II-13 인터뷰 질의서 작성 사례

인터뷰 결과

인터뷰 결과서			
회의일시	2012-07-06 / 13:30~14:30	회의장소	17층 회의실
회의명	기능 요구사항 인터뷰: 구매계약	작성자	OOO
참석자	OOO 팀장, OOO 책임, OOO 책임, OOO 운영자		

회의 논의 사항 기술

1. 주제
   - OOO 종합정보시스템 기능 요구사항 인터뷰: 구매계약

2. 내용
   - OO 기관의 구매계약에 대하여
     구매계약의 ISP수립, ISP 수립 당시 구매계약을 담당하던 담당자는 휴직상태
     구매계약을 담당하시는 OOO 책임님은 ISP 수립 당시 회계를 담당

   - 인지세법: 시스템 기능요구사항에 인지세법이 누락되어 있음
     전자계약 시스템이 현재 구축되어 있으나 일부만 구축되어 있음
     아직 협의된 사항이 아니나 후에 시스템 상에 구축이 확정 될 시 추가적으로 구축할 수 있도록 환경이 마련되어야 함
     시스템 기능의 프로세스 간 기능 연결
     기존의 문서만 작성하는 기능을 이제는 데이터로 관리해야 함
     원천문서 보기 기능 추가 및 시스템 상의 기능이 자동적으로 연결되고 전자결재의 연결이 유연하도록 각 기능의 연계를 잘 반영해야 함
     전자결제의 연동을 각 기능에 상세히 기술해야 함

   - 단가구매(소액구매): 단가구매의 경우 절차가 다름
     구매요구서를 작성하면서 발의가 잡히는 프로세스임
     단가구매의 경우 계약서가 없음

그림 II-14 인터뷰 결과서 작성 사례

Joint Application Development (JAD)

- Joint user-analyst meeting hosted by a facilitator
  - 10 to 20 users
  - 1 to 2 scribes as needed to record the session
  - Usually in a specially prepared room
- Meetings can be held electronically and anonymously
  - Reduces problems in group settings
  - Can be held remotely
- Sessions require careful planning to be successful
  - Users may need to bring documents or user manuals
  - Ground rules should be established

Questionnaires

- A set of written questions used to obtain information from individuals
- May be paper based or electronic (e.g., web based)
- Common uses:
  - Large numbers of people
  - Need both information and opinions
  - When designing for use outside the organization (customers, vendors, etc.)
- Typical response rates: < 50% (paper); < 30% (Web)

Questionnaire Steps

- Select the participants
  - Identify the population
  - Use representative samples for large populations
- Designing the questionnaire
  - Careful question selection
  - Remove ambiguities
- Administering the questionnaire
  - Working to get good response rate
  - Offer an incentive (e.g., a free pen)
- Questionnaire follow-up
  - Send results to participants
  - Send a thank-you

Document Analysis

- Provides information about the “as-is” system
- Review technical documents when available
- Review typical user documents:
  - Forms
  - Reports
  - Policy manuals
- Look for user additions to forms
- Look for unused form elements

Observation

- Users/managers often don't remember everything they do
- Checks validity of information gathered in other ways
- Behaviors may change when people are watched
  - Workers tend to be very careful when watched
  - Keep a low profile
  - Try not to interrupt or influence workers
- Be careful not to ignore periodic activities
  - Weekly … Monthly … Annually

Requirements-Gathering Techniques Compared

- A combination of techniques may be used
- Document analysis & observation require little training; JAD sessions can be very challenging

	Interviews	Joint Application Design	Questionnaires	Document Analysis	Observation
Type of information	As-is, improvements, to-be	As-is, improvements, to-be	As-is, improvements	As-is	As-is
Depth of information	High	High	Medium	Low	Low
Breadth of information	Low	Medium	High	High	Low
Integration of information	Low	High	Low	Low	Low
User involvement	Medium	High	Low	Low	Low
Cost	Medium	Low–Medium	Low	Low	Low to Medium

Alternative Techniques

- Concept Maps
  - Represent meaningful relationships between concepts
  - Focus individuals on a small number of key ideas
- User Stories, Story Cards & Task Lists
  - Associated with agile development methods
  - Very low tech, high touch, easily updatable, and very portable
  - Captured using story cards (index cards)
  - Capture both functional and nonfunctional requirements.

The System Proposal

- Combines all material created in planning & analysis
- Included sections:
  - Executive summary
    - Provides all critical information is summary form
    - Helps busy executives determine which sections they need to read in more detail
  - The system request
  - The workplan
  - The feasibility analysis
  - The requirements definition
  - Current models of the system (expected to evolve)

System Proposal Template

1. **Table of Contents**
2. **Executive Summary**

   A summary of all the essential information in the proposal so a busy executive can read it quickly and decide what parts of the proposal to read in more depth.
3. **System Request**

   The revised system request form (see Chapter 2).
4. **Workplan**

   The original workplan, revised after having completed analysis (see Chapter 2).
5. **Feasibility Analysis**

   A revised feasibility analysis, using the information from analysis (see Chapter 2).
6. **Requirements Definition**

   A list of the functional and nonfunctional business requirements for the system (this chapter).
7. **Functional Model**

   An activity diagram, a set of use case descriptions, and a use-case diagram that illustrate the basic processes or external functionality that the system needs to support (see Chapter 4).
8. **Structural Models**

   A set of CRC cards, class diagram, and object diagrams that describe the structural aspects of the to-be system (see Chapter 5). This may also include structural models of the current as-is system that will be replaced.
9. **Behavioral Models**

   A set of sequence diagrams, communication diagrams, behavioral-state machines, and a CRUDE matrix that describe the internal behavior of the to-be system (see Chapter 6). This may include behavioral models of the as-is system that will be replaced.
10. **Appendices**

    These contain additional material relevant to the proposal, often used to support the recommended system. This might include results of a questionnaire survey or interviews, industry reports and statistics, and so on.

Project Identification

- System Request
  - A document that describes the reasons for and the value added from building a new system
  - Contains the following elements:
    - Name of a project
    - Project sponsor
    - Business need
    - Business requirements
    - Business value: expected value that the system will provide
    - Special issues or constraints

DEVELOPING REQUIREMENTS

User Stories

Use Cases

UML

User Stories and Use Cases

- User Stories are short, simple descriptions of a feature told from the perspective of the person who wants the feature, usually a user or customer of the system. They typically follow a simple template:
- *As a <type of user>, I want <some goal>.*
- As a traveller, I want to book a hotel room.

Use-Cases: describing how the user will use the system

- A *use case* is a typical sequence of actions that a user performs in order to complete a given task
  - The objective of *use case analysis* is to model the system from the point of view of...
    - how users interact with this system
    - when trying to achieve their objectives

    It is one of the key activities in requirements analysis
  - A *use case model* consists of
    - a set of use cases
    - an optional description or diagram indicating how they are related

Actor actions	System responses
1. Click on a hotel logo	2. Display hotel details
3. Click “Book Now”	4. Display payment form
…	…

Use cases

- A use case should
  - Cover the *full sequence of steps* from the beginning of a task until the end.
  - Describe the *user's interaction* with the system ...
    - Not the computations the system performs.
  - Be written so as to be as *independent* as possible from any particular user interface design.
  - Only include actions in which the actor interacts with the system.
    - Not actions a user does manually nor internal actions of the system

Example description of a use case

Use case: **Open file**

Related use cases:
Generalization of:
• Open file by typing name
• Open file by browsing

Steps:

Actor actions	**System responses**
1. Choose ‘Open…’ command	2. File open dialog appears
3. Specify filename	
4. Confirm selection	5. Dialog disappears

Example (continued)

Use case: **Open file by typing name**

Related use cases:
Specialization of: Open file

Steps:

Actor actions	**System responses**
1. Choose ‘Open…’ command	2. File open dialog appears
3a. Select text field	
3b. Type file name	
4. Click ‘Open’	5. Dialog disappears

Example (continued)

Use case: **Open file by browsing**

Related use cases:
Specialization of: Open file
Includes: Browse for file

Steps:

Actor actions	**System responses**
1. Choose ‘Open…’ command	2. File open dialog appears
3. Browse for file	
4. Confirm selection	5. Dialog disappears

Example (continued)

Use case: **Attempt to open file that does not exist**

Related use cases:
Extension of: Open file by typing name

Actor actions	**System responses**
1. Choose ‘Open…’ command	2. File open dialog appears
3a. Select text field	
3b. Type file name	
4. Click ‘Open’	5. System indicates that file does not exist
6. Correct the file name	
7. Click ‘Open’	8 Dialog disappears