



Lecture 7: Requirements Elicitation

Kenneth M. Anderson
Object-Oriented Analysis and Design
CSCI 6448 - Spring Semester, 2002



Credit where Credit is Due

- Some material presented in this lecture is taken from section 3 of Maciaszek's "Requirements Analysis and System Design". © Addison Wesley, 2000

February 5, 2002

© Kenneth M. Anderson, 2002

2



Goals for this Lecture

- Introduce requirements elicitation
 - Cover the standard set of techniques used during this stage of the requirements phase
 - Also known as "Requirements Gathering"

February 5, 2002

© Kenneth M. Anderson, 2002

3



Topics

- Principles of Requirements Determination
- Requirements Elicitation
- Requirements Negotiation and Validation
- Requirements Management
- Problem Statements for Case Studies
- Requirements Business Model
- Requirements Document

February 5, 2002

© Kenneth M. Anderson, 2002

4

Principles of requirements determination

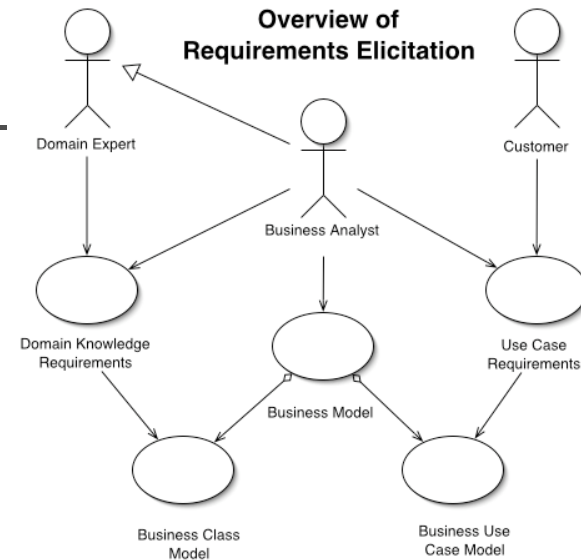
- Requirements define
 - System services
 - Function requirements
 - What must the system do?
 - Data requirements
 - What information must the system manage?
 - System constraints
 - Also known as Non-Functional Requirements
 - performance, security, robustness, etc.

February 5, 2002

© Kenneth M. Anderson, 2002

5

Overview of Requirements Elicitation



February 5, 2002

© Kenneth M. Anderson, 2002

6

Traditional methods of requirements elicitation

- Interviewing customers/domain experts
- Questionnaires
- Observation
- Study of (existing) documents and software systems

February 5, 2002

© Kenneth M. Anderson, 2002

7

Interviewing customers and domain experts

- Structured interview
 - Open-ended questions
 - Close-ended questions
- Unstructured interview
- Questions to be avoided
 - Opinionated questions
 - “Isn’t that an outdated way of doing things?”
 - Biased questions
 - “Are you going to upgrade to a better toolkit?”
 - Imposing questions
 - “Of course you already do configuration management, right?”

February 5, 2002

© Kenneth M. Anderson, 2002

8



Questionnaires

- In addition to interviews
- Close-ended questions
 - Multiple-choice, rating, and ranking questions
- Very difficult to “get right”
 - You need to know in advance what information you are looking for and how to ask questions that provide the information without biasing the answers
 - Need to account for unreturned surveys
 - e.g. if you distribute 100 surveys but only get 20 back, you can't assume you know the majority opinion



Observation

- This technique is drawn from anthropology
 - The analyst needs to understand the culture of the organization being observed
- Passive
 - analyst observes business activities without interruption or direct involvement (typically through video recordings)
- Active
 - analyst joins the team that is being observed
- Observations must occur for a prolonged period of time
 - to capture different types of activities and workloads
- Be aware that people tend to behave differently



Study of documents and software systems

- Use case requirements
 - Organizational documents
 - System forms and reports
 - these items can show how work is really done as opposed to how its specified
- Domain knowledge requirements
 - Domain journals and reference books
 - Proprietary software packages often contain a wealth of domain knowledge



Modern methods of requirements elicitation

- Prototyping
 - Already discussed previously in lecture
- Joint Application Development (JAD)
- Rapid Application Development (RAD)



Prototyping

- Throw-away prototype
 - Typically generated automatically by prototyping tools
 - Does not contain anything close to complete functionality
 - Used only to answer questions
- Evolutionary prototype
 - Used to build final product, but only when domain is well-understood and first version of system can be produced fairly quickly (but with incomplete functionality)



JAD

- The membership
 - Leader - moderator/domain expert
 - Scribe - records design decisions
 - Customers - they do the talking
 - Users
 - Managers
 - Developers - they do the listening
- Leverages “Oracle Effect”



RAD

- Evolutionary prototyping
- CASE tools with
 - code generation
 - round-trip engineering
- Specialists with Advanced Tools (SWAT)
 - the developers: best available, co-located with users
- Interactive JAD
 - SWAT Team takes place of Scribe
- Timeboxing
 - fixed development time; scope is trimmed back if team is running out of time



Requirements Negotiation/Validation

- After elicitation, requirements must be checked for inconsistencies, overlaps, and validity (in-scope);
- A priority must be placed on requirements as negotiated with the customer
- A requirements dependency matrix can aid this process

Requirements dependency matrix

Requirement	R1	R2	R3	R4
R1	X	X	X	X
R2	Conflict	X	X	X
R3			X	X
R4		Overlap	Overlap	X

Requirements management

- Requirements identification and classification
- Requirements hierarchies
- Change management
- Requirements traceability

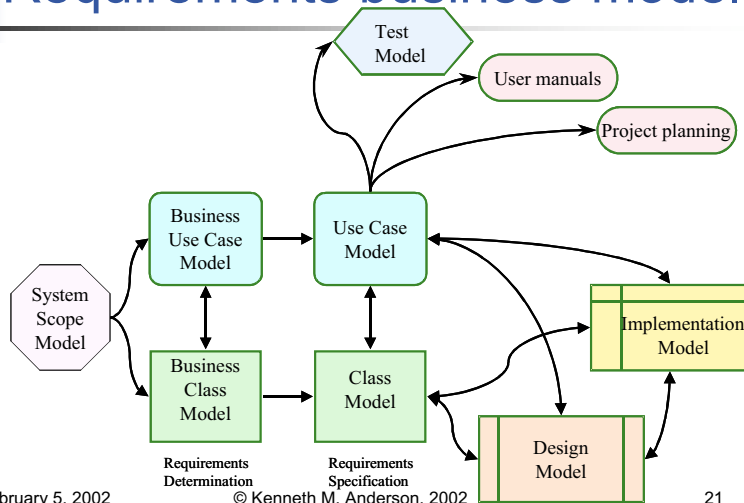
Requirements identification and classification

- Unique identifier
- Sequential number with document hierarchy
- Sequential number with requirement's category
- Database generated unique identifier
 - some databases can maintain referential integrity links that aid in requirements change management

Requirements hierarchies

- Requirements can be expressed in parent-child relationships similar to composition relationships
 - Children may be at a different level of abstraction
- 1. "The system shall schedule the next phone call to a customer upon telemarketer's request."
 - 1.1 "The system shall activate Next Call push button upon entry to Telemarketing Control form or when the previous call has terminated."
 - 1.2 "The system shall remove the call from the top of the queue of scheduled calls and make it the current call."
 - 1.3 etc.

Requirements business model



February 5, 2002

© Kenneth M. Anderson, 2002

21

Telemarketing

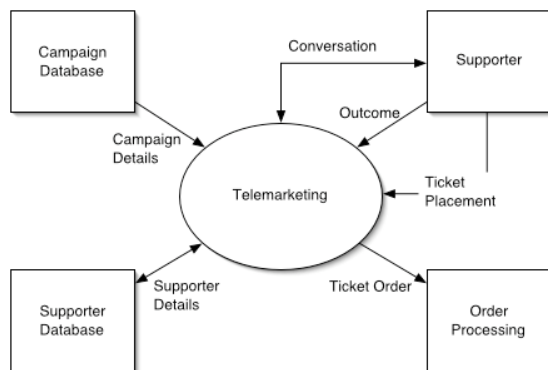
- Campaigns are started by the society trustees
- Campaigns have to be approved by the local government
- The design and planning of campaigns is supported by a separate Campaign Database application system
- There is also a separate Supporter Database that stores and maintains information about all past and present supporters – used to select supporters to be contacted in a particular campaign
- Orders from supporters for lottery tickets are recorded during telemarketing for perusal by the Order Processing system
- Order Processing System maintains status of orders in the Supporter Database

February 5, 2002

© Kenneth M. Anderson, 2002

22

System scope model

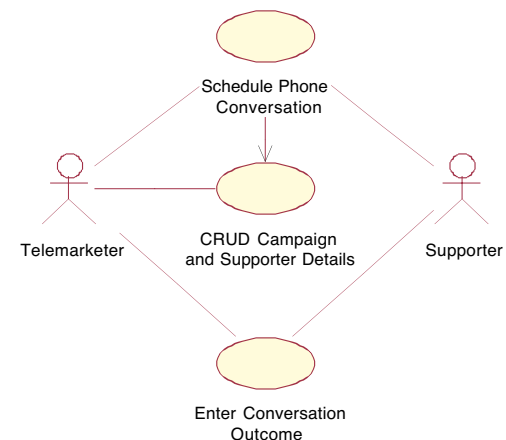


February 5, 2002

© Kenneth M. Anderson, 2002

23

Business use case model

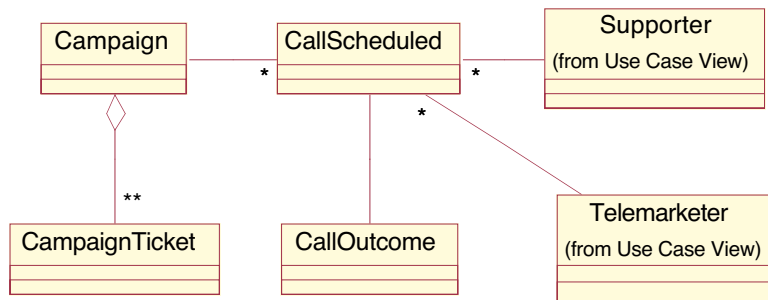


February 5, 2002

© Kenneth M. Anderson, 2002

24

Business class model



Requirements Document Table of Contents

- 1. Project Preliminaries**
 - 1.1 Purpose and Scope of the Product
 - 1.2 Business Context
 - 1.3 Stakeholders
 - 1.4 Ideas for Solutions
 - 1.5 Document Overview
- 2. System Services**
 - 2.1 The Scope of the System
 - 2.2 Function Requirements
 - 2.3 Data Requirements
- 3. System Constraints**
 - 3.1 Interface Requirements
 - 3.2 Performance Requirements
 - 3.3 Security Requirements
 - 3.4 Operational Requirements
 - 3.5 Political and Legal Requirements
 - 3.6 Other Constraints
- 4. Project Matters**
 - 4.1 Open Issues
 - 4.2 Preliminary Schedule
 - 4.3 Preliminary Budget

Appendices

- Glossary
- Business Documents and Forms
- References

Project preliminaries chapter

- Targets managers and decision makers
- Begins with purpose and scope of the project
- Makes a business case for the system
- Identifies stakeholders
- Offers initial ideas for the solution
- Includes an overview of the rest of the document

System services chapter

- Dedicated to the definition of system services
 - what the system must accomplish
- Likely to account for more than half of the entire document
- Contains high-level requirements business models
 - Context diagram (the system scope)
 - Business use case diagram (function requirements)
 - Business class diagram (data requirements)



System constraints chapter

- Dedicated to the definition of system constraints - how the system is constrained when accomplishing services with regard to
 - Interface requirements
 - Performance requirements
 - Security requirements
 - Operational requirements
 - Political and legal requirements
 - Other constraints
 - Usability
 - Maintainability



Project matters chapter

- Open issues
 - Future requirements
 - Current requirements to be implemented in the future – enhancements
 - Potential problems once when the system deployed
- Preliminary schedule
 - Human and other resources
 - Planning charts (PERT, Gantt)
- Preliminary budget
 - Project cost – range rather than figure



Appendices chapter

- Glossary
 - Terms
 - Acronyms
 - Abbreviations
- Documents and forms
 - Examples of completed (filled in) forms
- References
 - To books and other published sources
 - Meetings' minutes, memoranda, internal documents



Summary

- Requirements determination is about discovering requirements and documenting them
- Two lines of discovery – the discovery from the domain knowledge and from the use cases
- Methods of requirements elicitation include interviewing customers and domain experts, questionnaires, observation, study of documents and software systems, prototyping, JAD and RAD
- Requirements negotiation and validation to resolve overlaps and conflicts
- Requirements have to be managed
- Requirements business model uses diagrams – Context Diagram, Business Use Case Diagram, and Business Class Diagram
- The resulting document is called the Requirements Document