ISYS2110

Analysis and Design of Web Information Systems

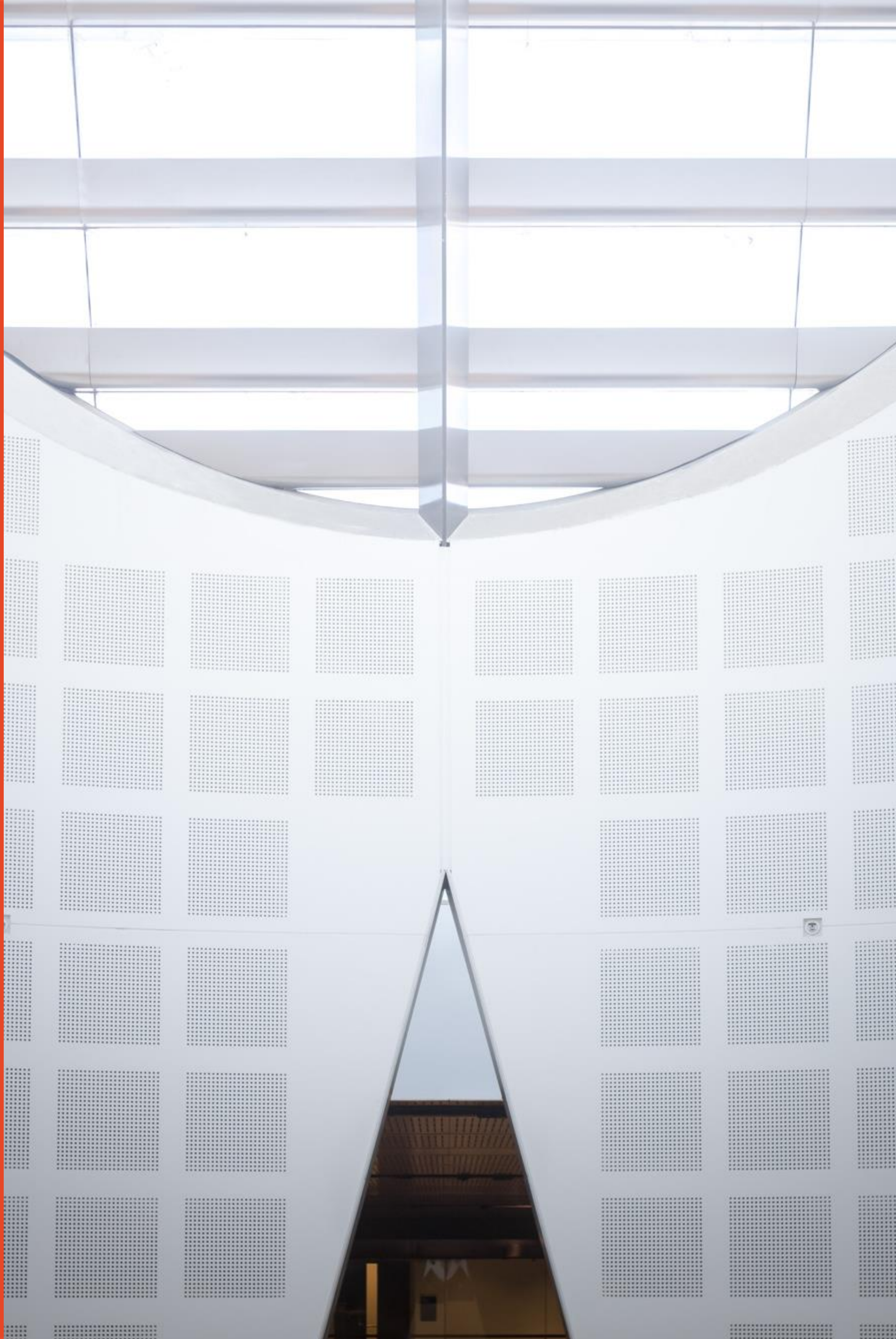
Lecture 5

Requirements Modeling

Semester 1, 2018

Dr Rabiul Hasan

THE UNIVERSITY OF SYDNEY



Source: Scott Tilley & Harry J. Rosenblatt's Systems Analysis And Design, 11th Ed © 2017 by Cengage Learning US

Recapture From Lecture 3

What we have covered on the topic:

Managing Systems Project

- Scope, Time, Cost
- WBS, Gantt Chart, CPM, PERT
- Quality, Conflict, and Risk

What Will We Do Today ?

- Lecture
  - Requirement Prioritization
  - Why, what, who issues
  - UML diagrams
  - Team-based approaches – JAD, RAD, Agile
  - Modeling tools – FDD, DFD, Use Case Diagram, Sequence Diagram

- Class activities
  - Critical Thinking / No Problem Solving Today
  - https://padlet.com
  - https://answergarden.ch

- Tutorial:
- Assessment
  - Assignment 1: Due this Sunday
  - Quiz1: Due in week 6, Friday, during the lecture

- Announcement (if any):

Learning Objectives

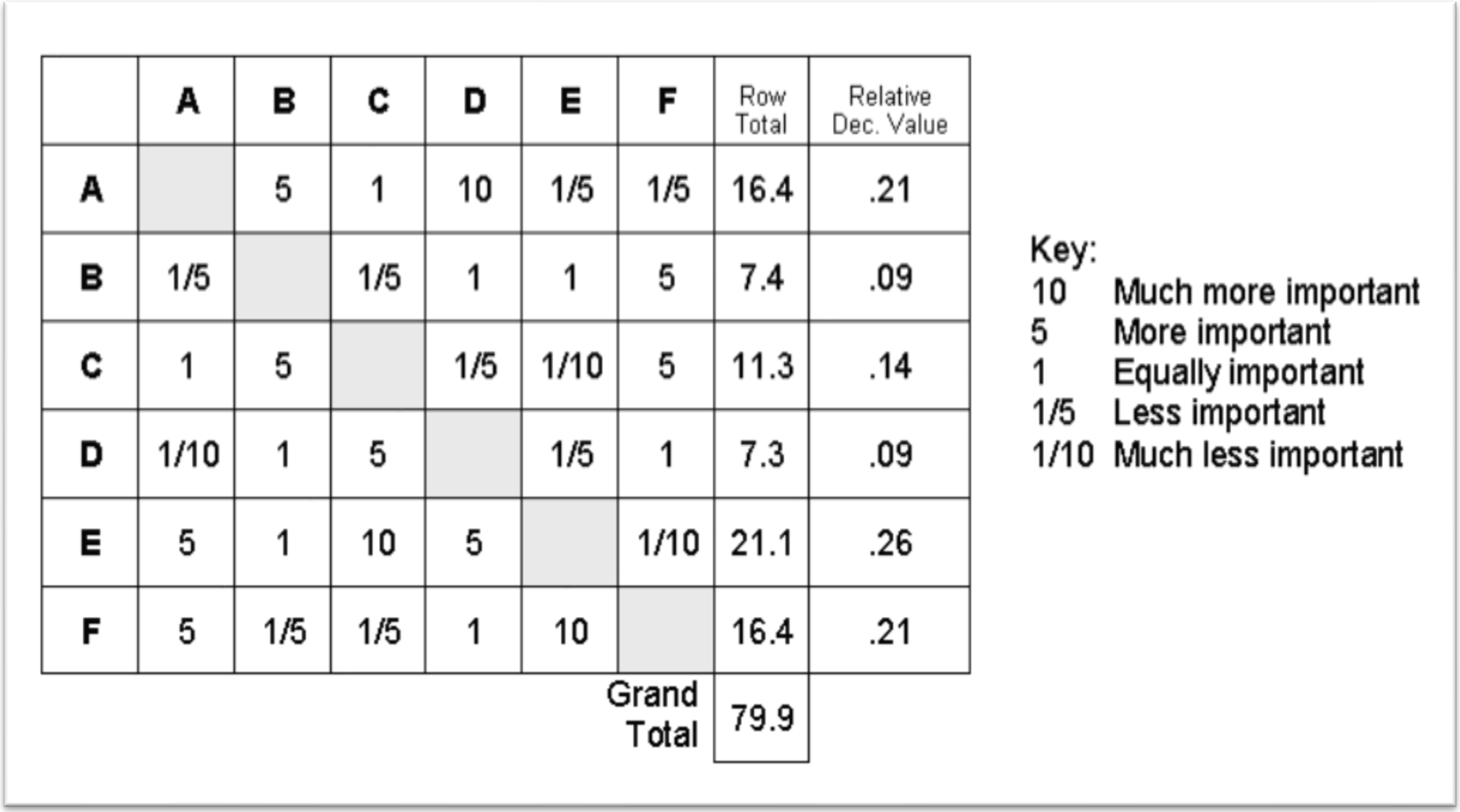
- Understand the process of computing requirements prioritization
- Describe requirements modeling as the first part of systems analysis activities

- Explain Team-based techniques (JAD, RAD, Agile) for modeling requirements
- Use a functional decomposition diagram (FDD) to model business functions and processes

- Describe the Unified Modeling Language (UML) and examples of UML diagrams
- List and describe system requirements, including outputs, inputs, processes, performance, and controls

Computing Requirements Prioritization

-- L-shaped Matrix



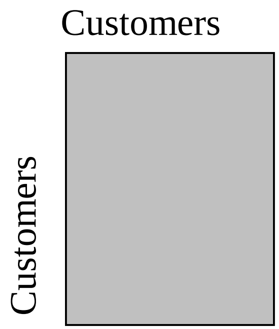
	A	B	C	D	E	F	Row Total	Relative Dec. Value
A		5	1	10	1/5	1/5	16.4	.21
B	1/5		1/5	1	1	5	7.4	.09
C	1	5		1/5	1/10	5	11.3	.14
D	1/10	1	5		1/5	1	7.3	.09
E	5	1	10	5		1/10	21.1	.26
F	5	1/5	1/5	1	10		16.4	.21
						Grand Total	79.9	

Key:

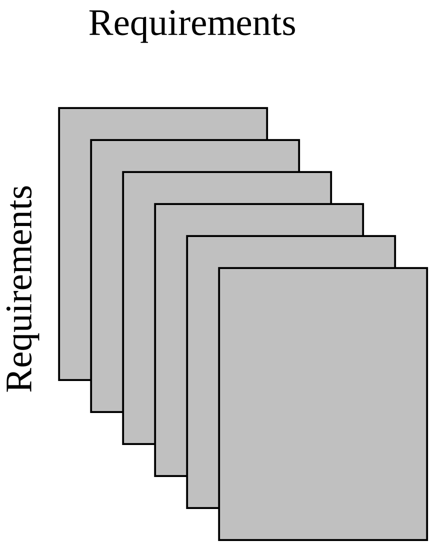
10	Much more important
5	More important
1	Equally important
1/5	Less important
1/10	Much less important

Adapted from: Rose, K. 2014

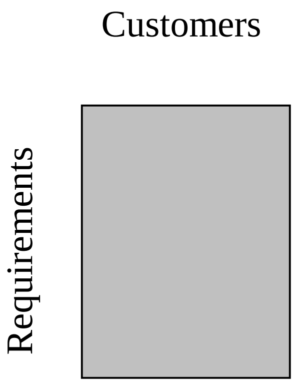
Computing Requirements Prioritization
-- L-shaped Matrix



Prepare a customer/stakeholders matrix to get customer values



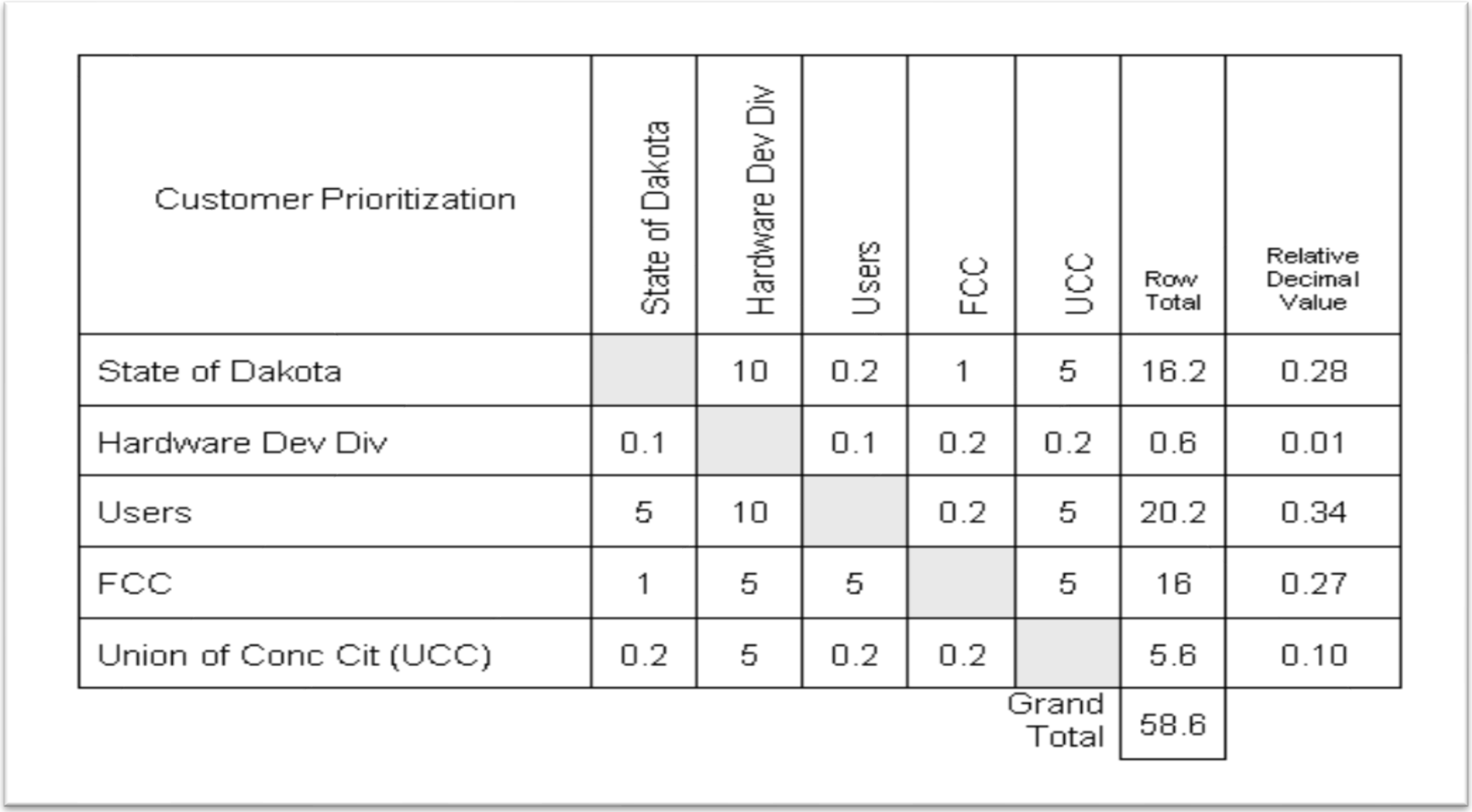
Prepare a requirements matrix for each customer / stakeholders to get requirement values



Combine customer /stakeholder and requirement matrices to get overall values for requirements

Adapted from: Rose, K. 2014

Computing Requirements Prioritization
-- Customer Prioritization Matrix



Customer Prioritization	State of Dakota	Hardware Dev Div	Users	FCC	UCC	Row Total	Relative Decimal Value
State of Dakota		10	0.2	1	5	16.2	0.28
Hardware Dev Div	0.1		0.1	0.2	0.2	0.6	0.01
Users	5	10		0.2	5	20.2	0.34
FCC	1	5	5		5	16	0.27
Union of Conc Cit (UCC)	0.2	5	0.2	0.2		5.6	0.10
					Grand Total	58.6	

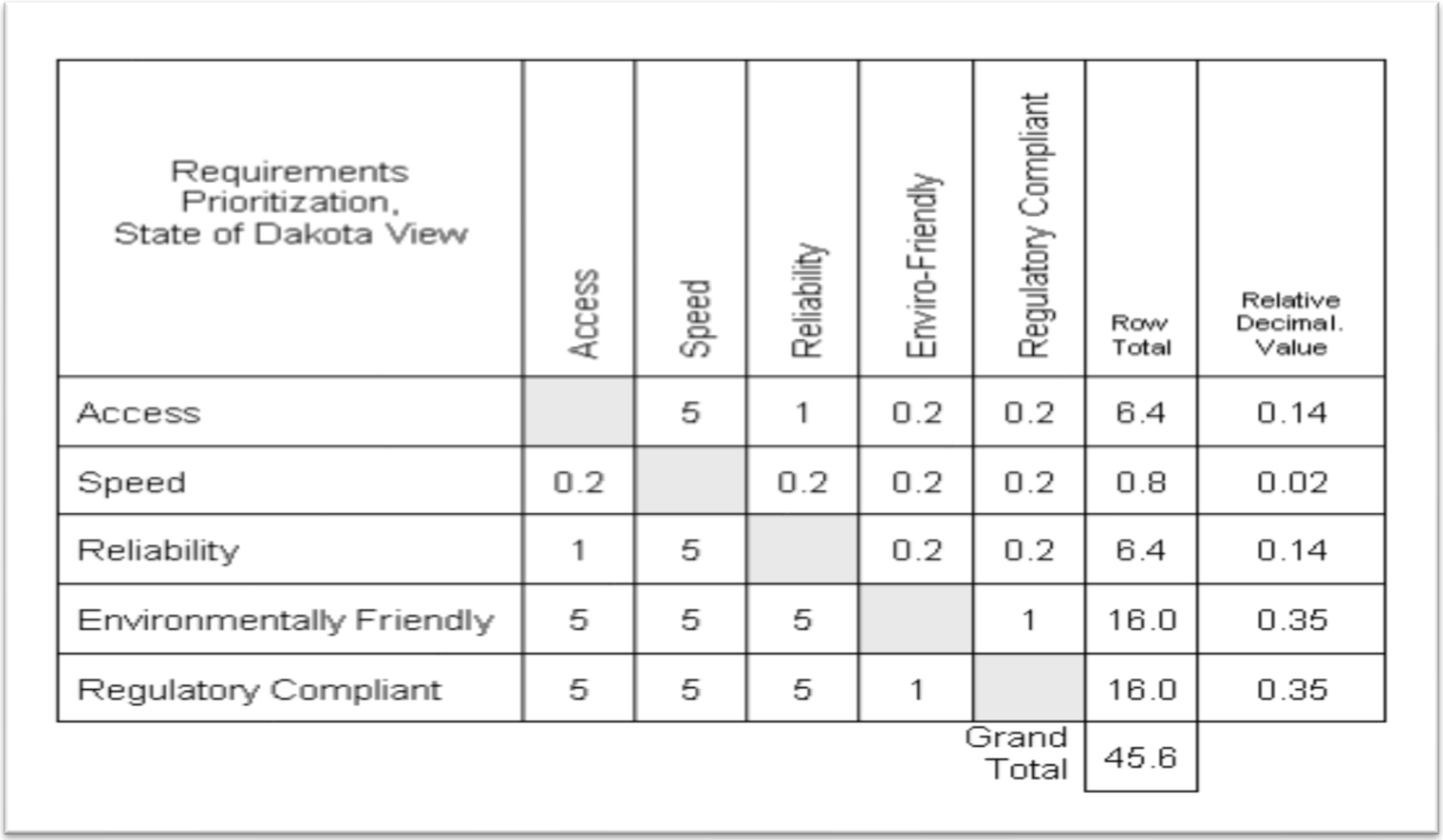
Adapted from: Rose, K. 2014

Computing Requirements Prioritization
-- Requirements List

- Requirements list needs to be unique based on identified stakeholders. For instance, identified requirements includes the following:
  - Access
  - Speed
  - Reliability
  - Environmentally friendly
  - Regulatory compliant

Adapted from: Rose, K. 2014

Computing Requirements Prioritization



Requirements Prioritization, State of Dakota View	Access	Speed	Reliability	Enviro-Friendly	Regulatory Compliant	Row Total	Relative Decimal. Value
Access		5	1	0.2	0.2	6.4	0.14
Speed	0.2		0.2	0.2	0.2	0.8	0.02
Reliability	1	5		0.2	0.2	6.4	0.14
Environmentally Friendly	5	5	5		1	16.0	0.35
Regulatory Compliant	5	5	5	1		16.0	0.35
					Grand Total	45.6	

Adapted from: Rose, K. 2014

Computing Requirements Prioritization

- You need to develop Requirements Prioritization tables for each of these customers or stakeholders:
  - **Hardware Development Division View** (Relative Decimal Value: 0.32, 0.21, 0.32, 0.01, 0.15 for access, speed, reliability, environmentally friendly, and regulatory compliant)
  - **Users View** (0.31, 0.16, 0.31, 0.02, 0.20)
  - **Federal Communications Commission (FCC) View** (0.06, 0.06, 0.06, 0.06, 0.76)
  - **Union Concerned Citizens (UCC) View** (0.04, 0.04, 0.04, 0.61, 0.27)

Adapted from: Rose, K. 2014

Computing Requirements Prioritization

Customer-Customer Weighting Matrix

Customer Prioritzation	State of Dakota	Hardware Dev Div	Users	FCC	UCC	Row Total	Relative Decimal Value
State of Dakota		10	0.2	1	5	16.2	**0.28**
Hardware Dev Div	0.1		0.1	0.2	0.2	0.6	0.01

Requirement-Requirement Weighting Matrix

Requirements Prioritization, State of Dakota View	Access	Speed	Reliability	Enviro-Friendly	Regulatory Compliant	Row Total	Relative Decimal Value
Access		5	1	0.2	0.2	6.4	**0.14**
Speed	0.2		0.2	0.2	0.2	0.8	0.02

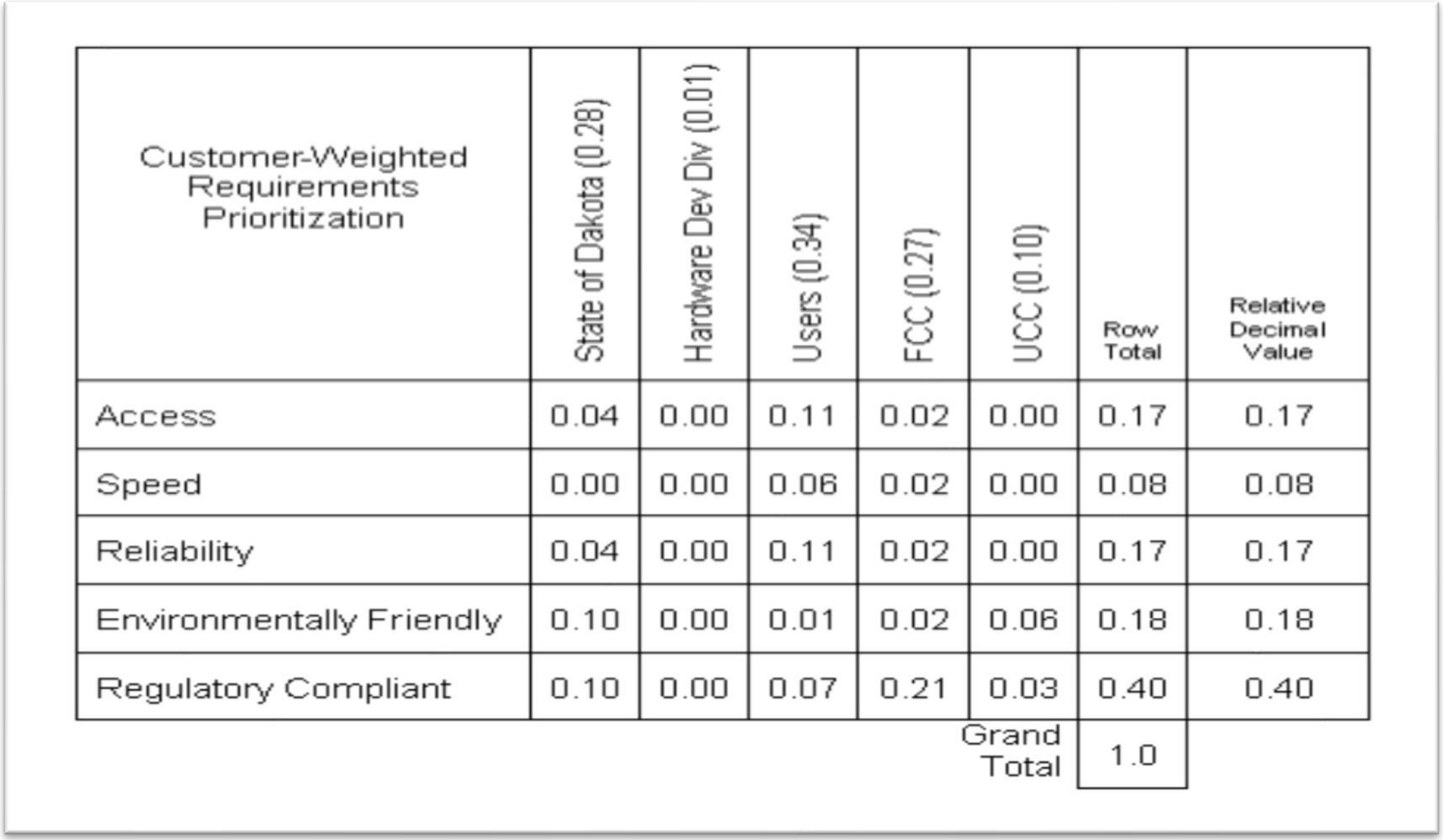
.28 x .14 = .04

Requirement-Customer Priorities Matrix

Customer-Weighted Requirements Prioritization	State of Dakota (.28)	Hardware Dev Div (.01)	Users (.34)	FCC (.27)	UCC (.10)	Row Total	Relative Decimal Value
Access	0.04	0.00	0.11	0.02	0.00	0.17	0.17
Speed	0.00	0.00	0.06	0.02	0.00	0.08	0.08

Adapted from: Rose, K. 2014

Computing Requirements Prioritization



Customer-Weighted Requirements Prioritization	State of Dakota (0.28)	Hardware Dev Div (0.01)	Users (0.34)	FCC (0.27)	UCC (0.10)	Row Total	Relative Decimal Value
Access	0.04	0.00	0.11	0.02	0.00	0.17	0.17
Speed	0.00	0.00	0.06	0.02	0.00	0.08	0.08
Reliability	0.04	0.00	0.11	0.02	0.00	0.17	0.17
Environmentally Friendly	0.10	0.00	0.01	0.02	0.06	0.18	0.18
Regulatory Compliant	0.10	0.00	0.07	0.21	0.03	0.40	0.40
					Grand Total	1.0	

Adapted from: Rose, K. 2014

Importance of Requirements Modeling

- Specifications are extremely long. Some documents can contain hundreds of pages and thousands of requirements. And for sure, we all experience deficit of time at work.

- Specifications are complex. Reading a technical specification is not the same as reading Harry Potter's adventures. You need to read once, then read one more time, realize what's written and think about it.

- Visualization matters

Adapted from Liza, 2017

Requirements Modeling –Skills ?

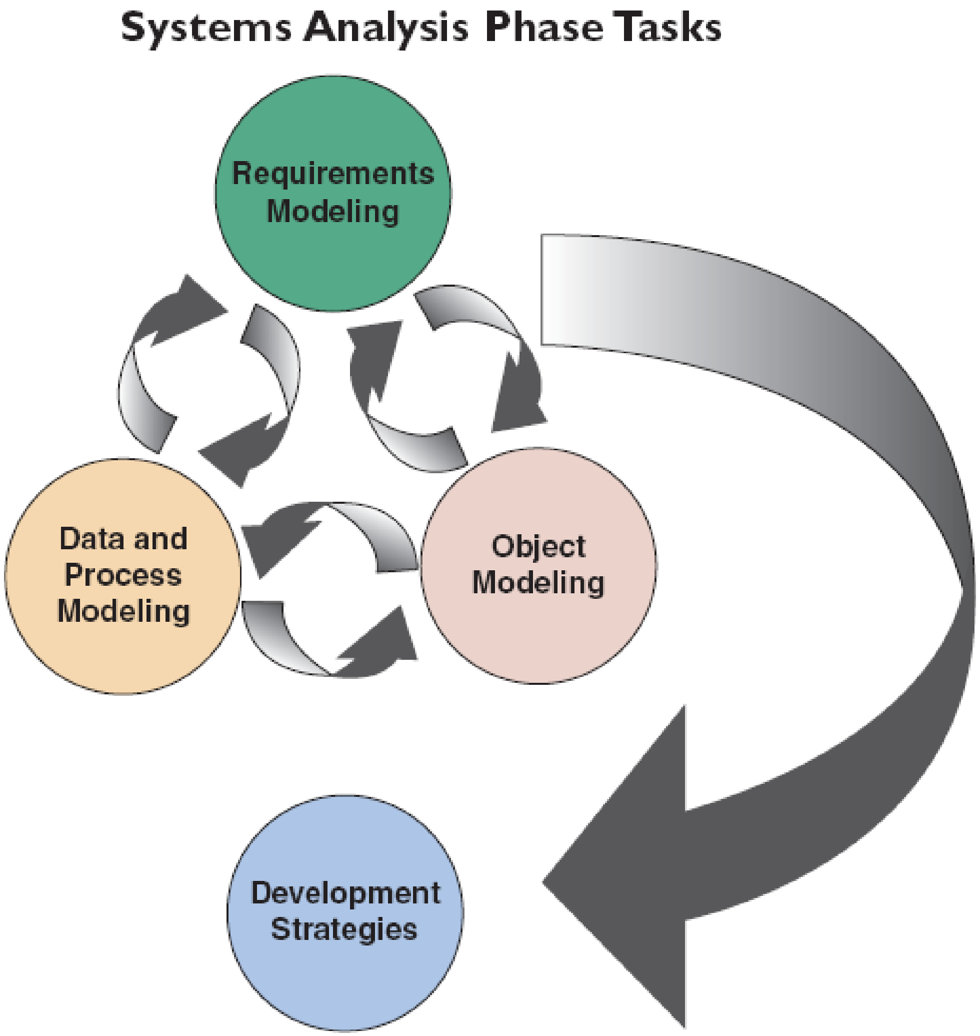
- Strong analytical and interpersonal skills needed

Critical Thinking: Exercise 1

- Why are analytical and interpersonal skills important when modeling requirements of a new system?

Requirements Modeling –
The First Part of Systems Analysis Activities

- The systems analysis phase consists of requirements modeling, data and process modeling, object modeling, and consideration of development strategies.
- Notice that the systems analysis tasks are interactive



Systems Analysis Activities

- **Requirements modeling**
  - Involves identification of the requirements for new system
  - Includes – why, what, who issues
- Data and process modeling
  - Graphically represents system data and processes
- Object modeling
  - Involves creation of objects to represent people, things, transactions, and events
- Development strategies
  - Include software trends, development alternatives, and outsourcing

Requirements Modeling

- Objectives of Requirements Modeling:
  - **Discover the boundaries** of the new system (or software) and how it must interact with its environment within the new problem domain
  - **Detect and resolve conflicts** between (user) requirements
  - **Negotiate priorities of stakeholders**
  - **Prioritize requirements**

  - **Elaborate system requirements**, defined in the requirement specification document, such that managers can give realistic project estimates and developers can design, implement, and test
  - **Classify requirements** information into various categories and allocate requirements to sub-systems
  - **Evaluate requirements** for desirable qualities

Gregor and colleagues, 2010

Requirements Modeling

- This is an essential task in specifying requirements
- Map elements obtained by elicitation to a more precise form
- Help better understand the problem
- Help find what is missing or needs further discussion

Gregor and colleagues, 2010

Requirements Modeling – Basic Questions

- In lecture 2, we have discussed how to investigate and collect requirements of a new system. There are still many questions to be clarified:
  - How to decide what is important and a priority?
  - How to ensure that nothing is forgotten?
  - How to validate that the findings of the analysis are good?
  - How to use models in specific context?
  - What aspects to model in terms of **Why-What-Who** ?

Gregor and colleagues, 2010

Requirements Modeling – "Why Issues"?

- The goals for a new system need to be identified, analyzed, and refined.
- Such goals are usually obtained by analyzing problems with the current situation, identifying new opportunities, exploring scenarios of interaction, and so on.
- Beside functional goals (e.g., satisfaction of requests, information of the state of affairs) there are many non-functional ones (e.g., safety, security, performance, evolvability, etc.).

Lamsweerde, 2001; Reliable Software Technologies

Requirements Modeling – “What Issues”?

- The requirements operationalizing the various goals identified need to be defined precisely and related to each other; in parallel, the assumptions made in the operationalization process need to be made explicit and documented.

- Beside functional requirements about services to be provided there is a wide spectrum of non-functional requirements about quality of service.

Lamsweerde, 2001; Reliable Software Technologies

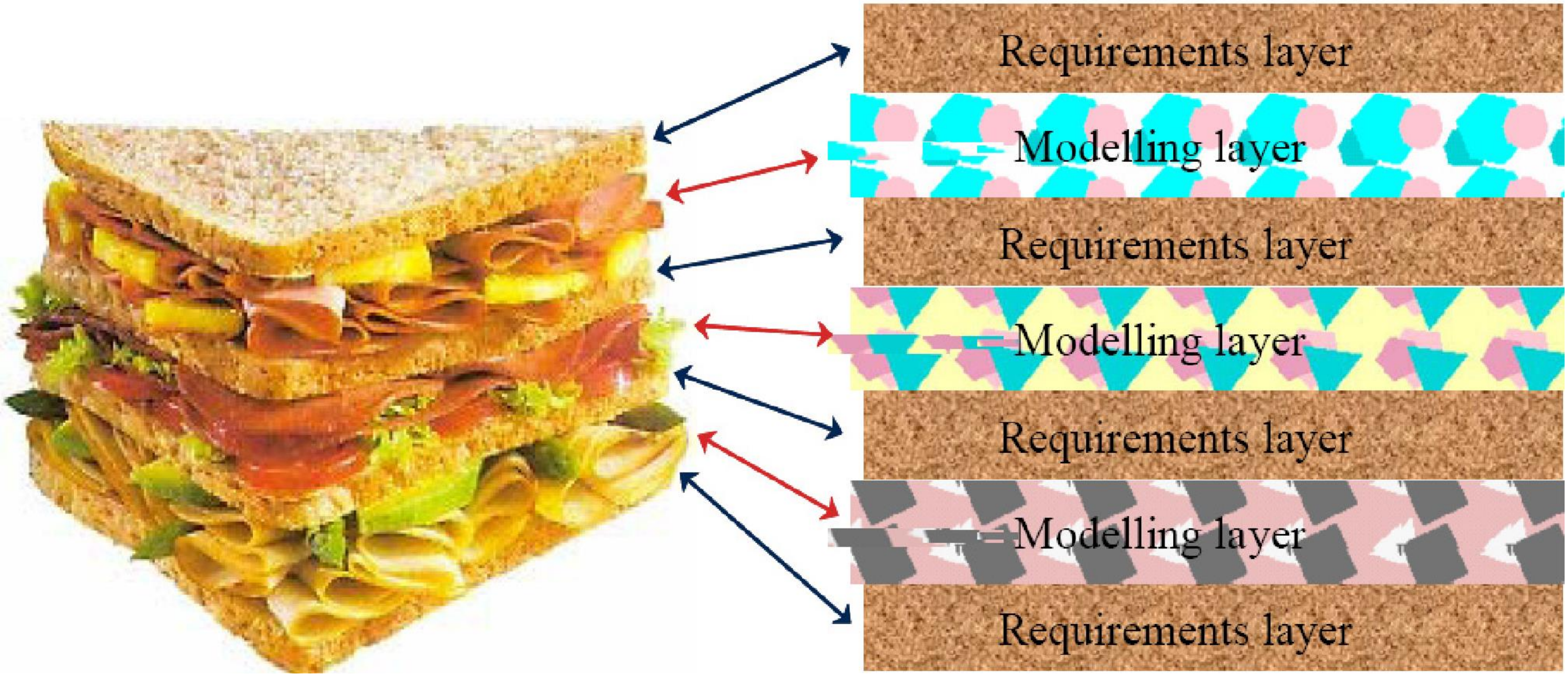
Requirements Modeling – "Who Issues"?

- The requirements need to be assigned as contractual obligations among the various agents forming the composite system-to-be.
- These include the software to be developed, human agents, sensors/component of a machine, existing software, etc.

Lamsweerde, 2001; Reliable Software Technologies

Requirements Modeling

- Elicitation/analysis and modeling are intermixed



Gregor and colleagues, 2010; adapted from www.telelogic.com

Requirements Modeling

- Informal: natural language
- Goal-oriented modeling (GRL)
- Functional modeling: UML (Unified Modeling Language) Diagrams

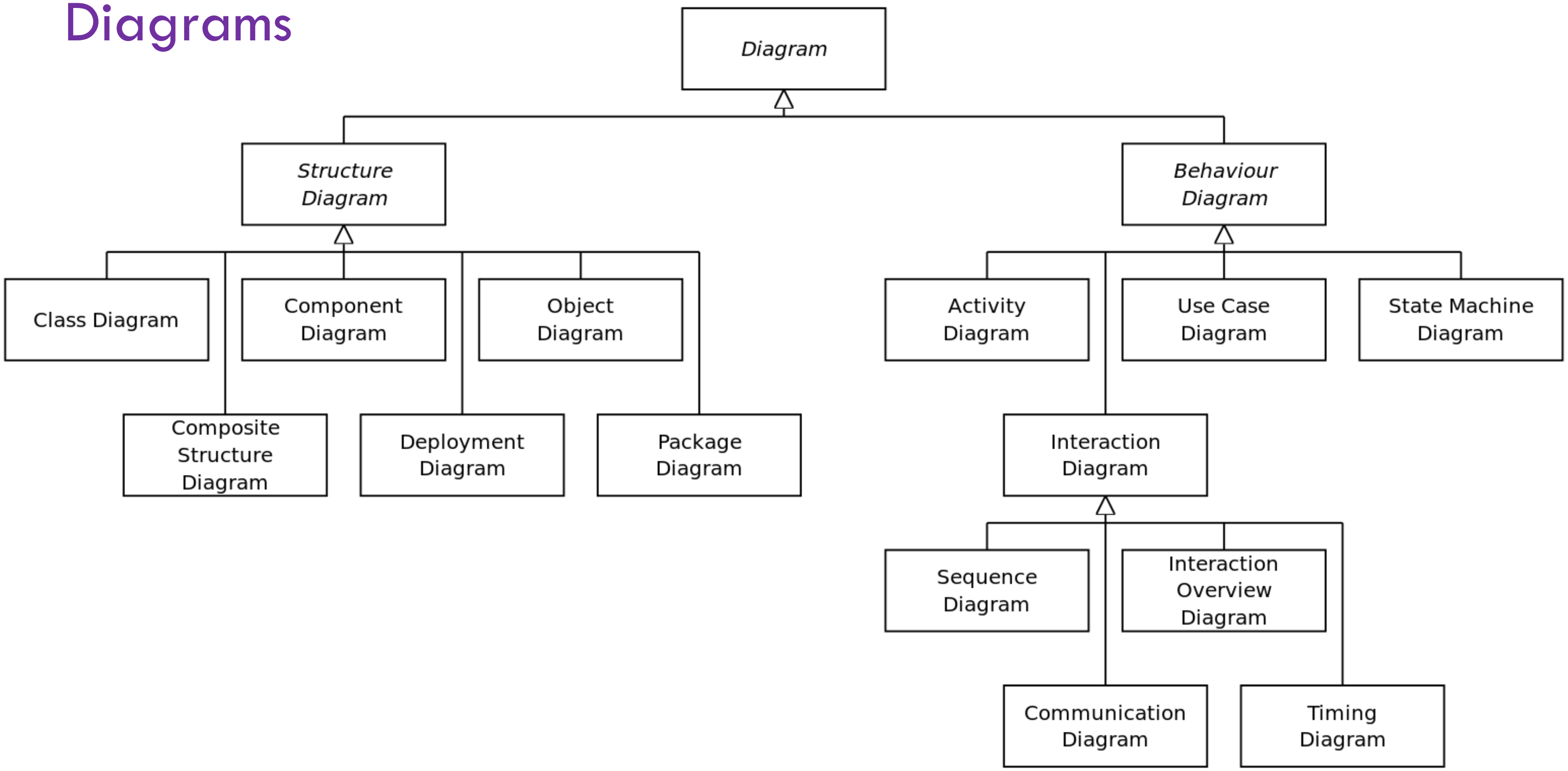


Image source: wikimedia.org

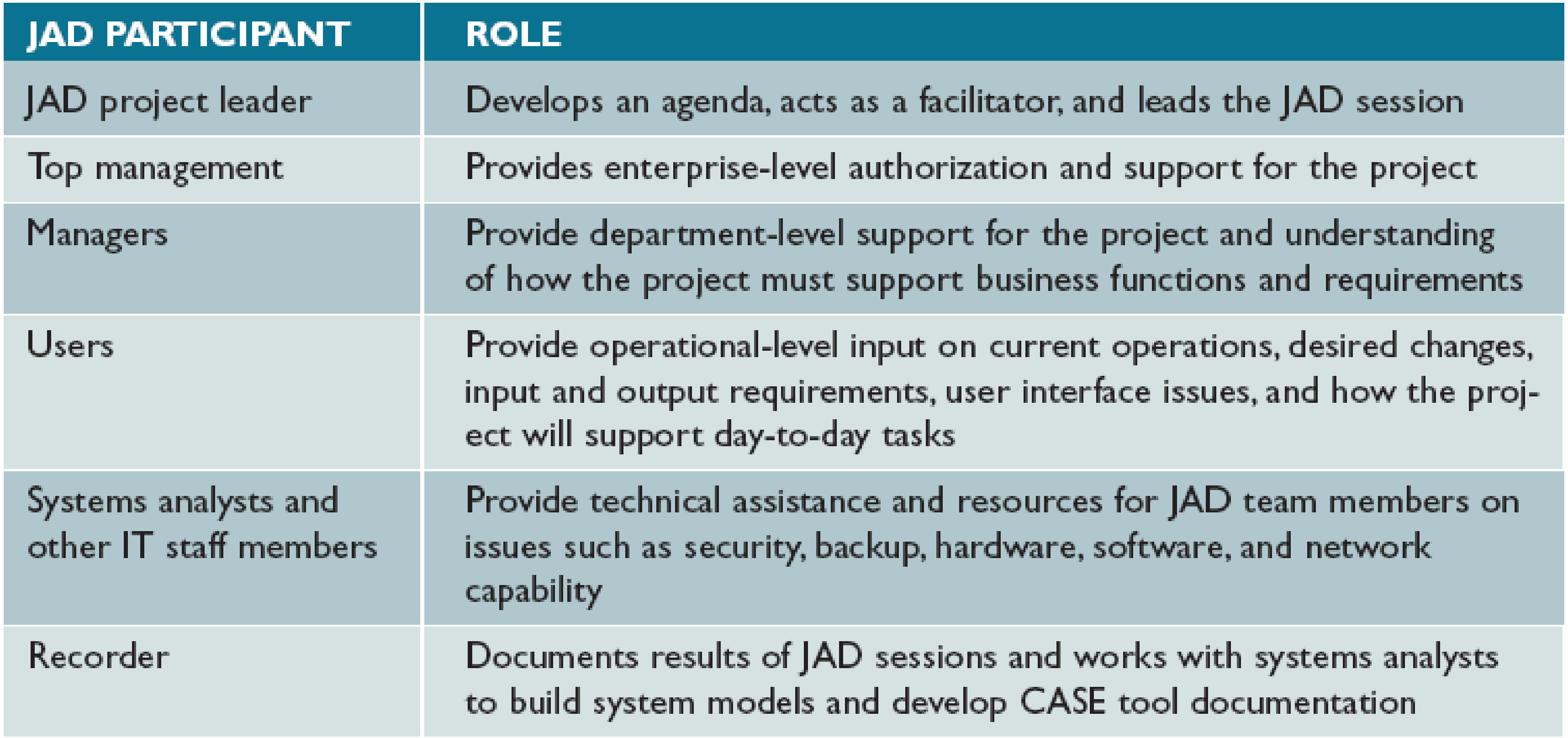
Team-Based Approaches for Requirements Modeling

- **Team-Based Approaches: JAD, RAD, and Agile Methods**
  - Goal - To deliver the best possible system at the lowest possible cost in the shortest possible time
    - Joint application development (JAD) brings users into the design process (briefly discussed in lecture 2)
    - Rapid application development (RAD) is a condensed version of the system development life cycle (briefly discussed in lecture 1)
    - Agile methods stress intense interaction between developers and users (discussed in lecture 3)

JAD Approach for Requirements Modeling

- Brings users into the development process as active participants
- **User Involvement** (formal or informal)
  - Helps create a successful system
- **JAD Participants and Roles**
  - Project leader and one or more members
  - Participants should be insulated from distractions of day-to-day operations

JAD Approach for Requirements Modeling

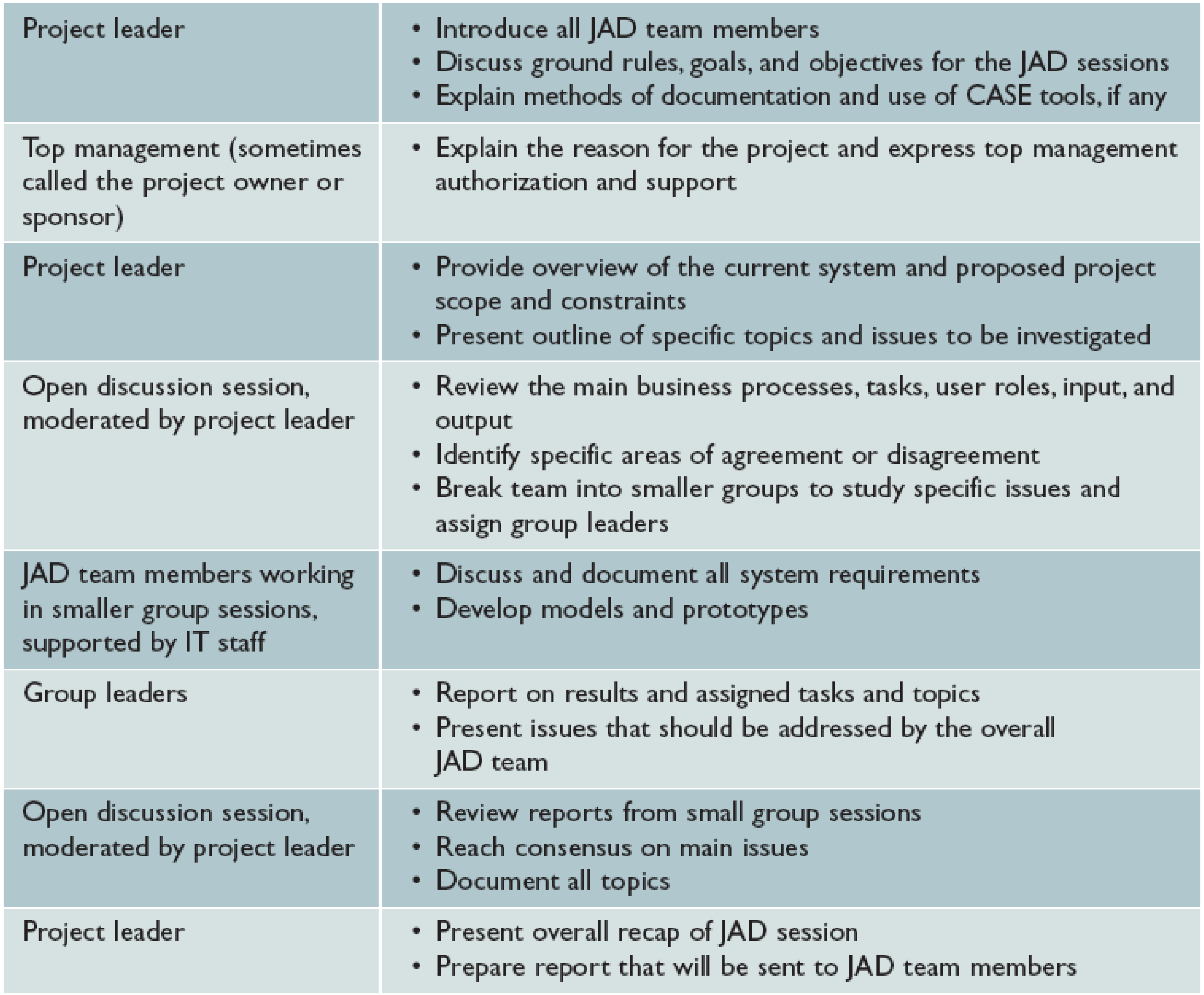


JAD PARTICIPANT	ROLE
JAD project leader	Develops an agenda, acts as a facilitator, and leads the JAD session
Top management	Provides enterprise-level authorization and support for the project
Managers	Provide department-level support for the project and understanding of how the project must support business functions and requirements
Users	Provide operational-level input on current operations, desired changes, input and output requirements, user interface issues, and how the project will support day-to-day tasks
Systems analysts and other IT staff members	Provide technical assistance and resources for JAD team members on issues such as security, backup, hardware, software, and network capability
Recorder	Documents results of JAD sessions and works with systems analysts to build system models and develop CASE tool documentation

Typical JAD participants and roles

JAD Approach for Requirements Modeling

Typical agenda for a JAD session



Project leader	• Introduce all JAD team members • Discuss ground rules, goals, and objectives for the JAD sessions • Explain methods of documentation and use of CASE tools, if any
Top management (sometimes called the project owner or sponsor)	• Explain the reason for the project and express top management authorization and support
Project leader	• Provide overview of the current system and proposed project scope and constraints • Present outline of specific topics and issues to be investigated
Open discussion session, moderated by project leader	• Review the main business processes, tasks, user roles, input, and output • Identify specific areas of agreement or disagreement • Break team into smaller groups to study specific issues and assign group leaders
JAD team members working in smaller group sessions, supported by IT staff	• Discuss and document all system requirements • Develop models and prototypes
Group leaders	• Report on results and assigned tasks and topics • Present issues that should be addressed by the overall JAD team
Open discussion session, moderated by project leader	• Review reports from small group sessions • Reach consensus on main issues • Document all topics
Project leader	• Present overall recap of JAD session • Prepare report that will be sent to JAD team members

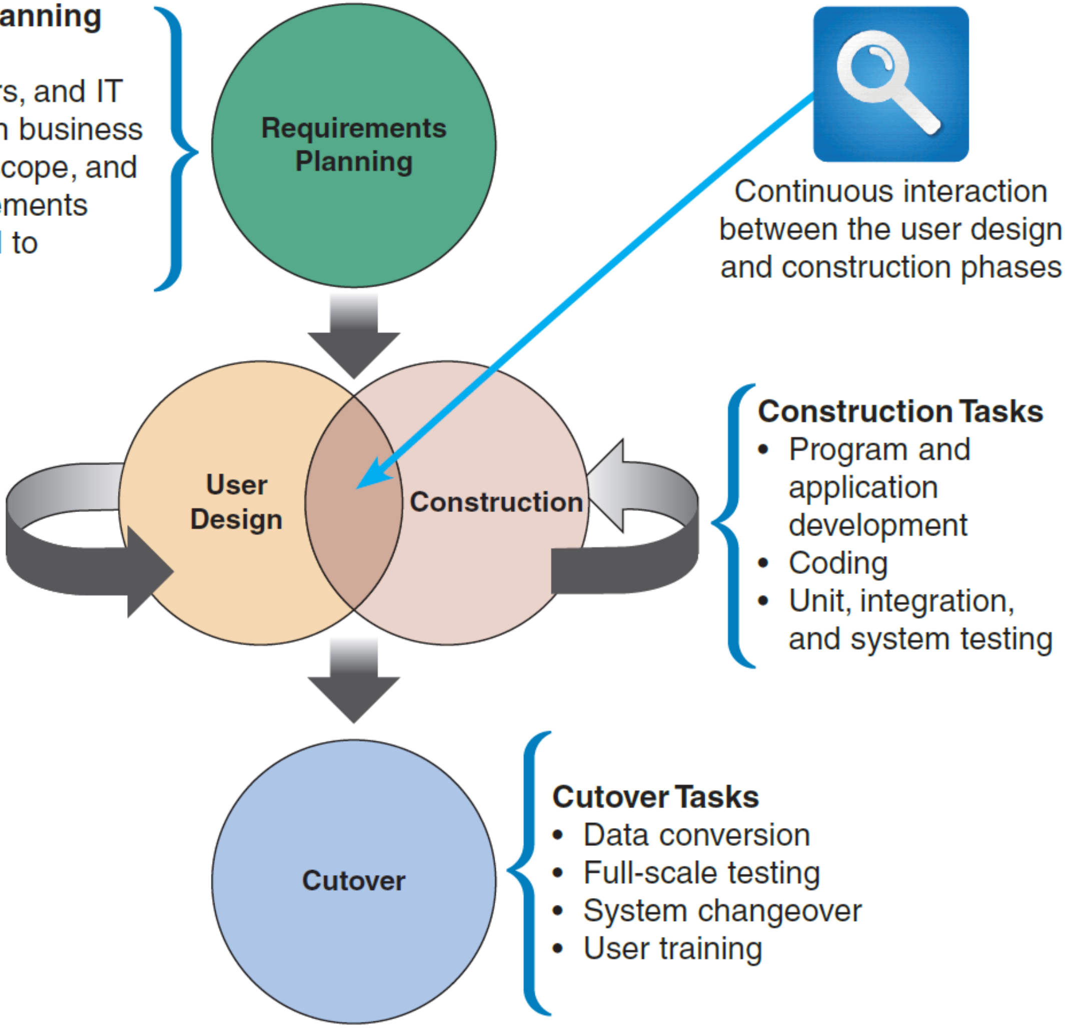
JAD Approach for Requirements Modeling

- **JAD Advantages and Disadvantages**
  - Disadvantages
    - More expensive than traditional methods
    - Can be cumbersome if the group is too large
  - Advantages
    - Allows key users to participate effectively
    - Users are more likely to feel a sense of ownership
    - Produces a more accurate statement of system requirements, a better understanding of common goals, and a stronger commitment to the success of the new system

JAD Approach for Requirements Modeling

- Uses a group approach like JAD
- End product - New information system
- Complete methodology
  - Includes a four-phase life cycle that parallels the traditional SDLC
  - Reduces cost and development time
  - Increases the probability of success
  - Relies on prototyping and user involvement
    - Prototypes are modified based on user input

RAD Approach for Requirements Modeling



Requirements Planning Tasks

- Users, managers, and IT staff agree upon business needs, project scope, and systems requirements
- Obtain approval to continue

User Design Tasks

- Interact with users
- Build models and prototypes
- Conduct intensive JAD-type sessions

Construction Tasks

- Program and application development
- Coding
- Unit, integration, and system testing

Cutover Tasks

- Data conversion
- Full-scale testing
- System changeover
- User training

Continuous interaction between the user design and construction phases

The four phases of the RAD model are requirements planning, user design, construction, and cutover. Notice the continuous interaction between the user design and construction phases.

RAD Approach for Requirements Modeling

- **RAD Objectives**
  - Cut development time and expense
    - Involve users in every phase of systems development
    - Must have the right IT resources, skills, and management support

- **RAD Advantages and Disadvantages**
  - Advantage – Helps develop systems quickly with significant cost savings
  - Disadvantages
    - Does not emphasize the company’s strategic business needs
    - Less time to develop quality, consistency, and design standards

Agile Approach and Requirements Modeling

- At present the majority of software projects are realised using the agile approach.

- The software companies aim to shorten time spent on information system developing so as to be more competitive and to lower project costs

Adapted from Knieżová, 2013

Agile Approach and Requirements Modeling

- Advantages
  - Very flexible and efficient in dealing with change
  - Frequent deliverables constantly validate the project and reduce risk

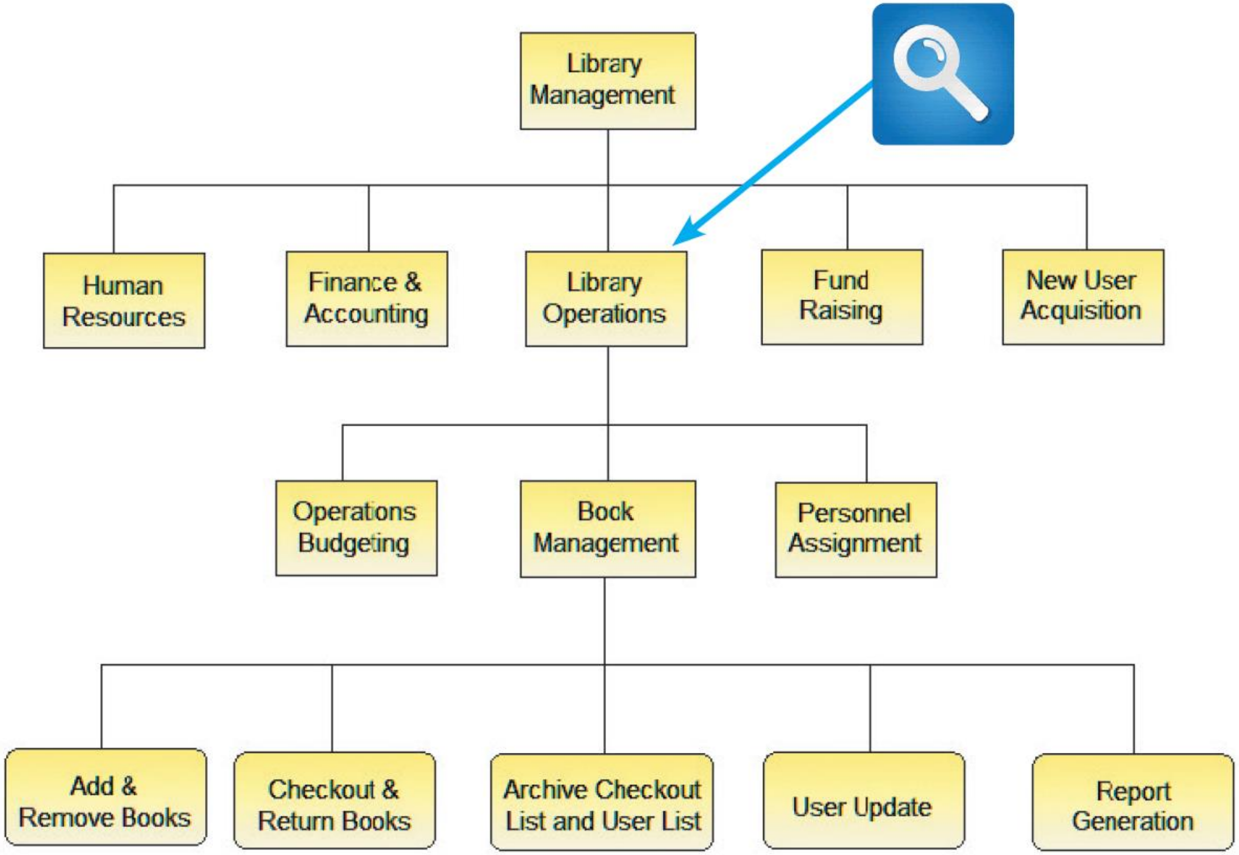
- Disadvantages
  - Team members need a high level of technical and interpersonal skills
  - Lack of structure and documentation can introduce risk factors
  - May be subject to significant change in scope

Modeling Tools and Techniques

- Involve graphical methods and nontechnical language that represent the system at various stages of development
- Systems analysts:
  - Build fact-finding results into models
  - Study the models to determine whether additional fact-finding is needed
- **Functional Decomposition Diagrams (FDD)**
  - Top-down representation of a function or process
  - Help analysts show business functions and how they are organized into lower-level processes

Modeling Tools and Techniques

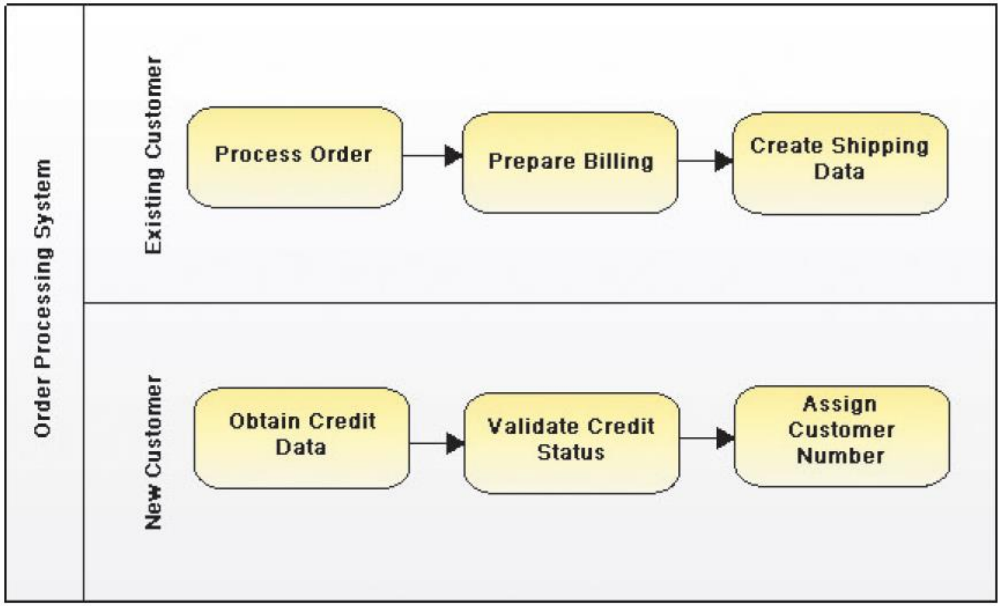
- Functional Decomposition Diagrams (FDD)



This Visible Analyst FDD shows a library system with five top-level functions. The Library Operations function includes two additional levels of processes and sub processes. Source: Visible Systems Corporation.

Modeling Tools and Techniques

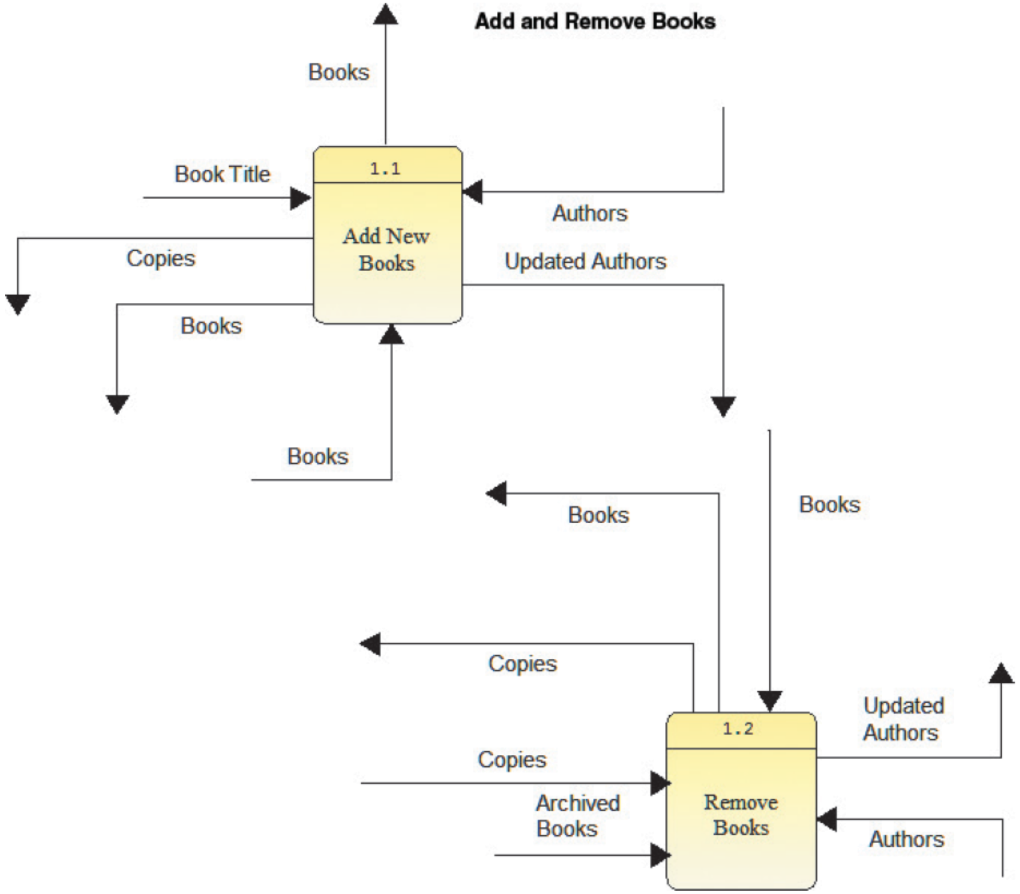
- **Business Process Modeling (BPM)**
  - Represents one or more business processes
  - Business process modeling notation (BPMN)
    - Models that use a standard language
    - Includes shapes and symbols to represent events, processes, and workflows



Using the Visible Analyst CASE tool, an analyst can create a business process diagram. The overall diagram is called a pool, and the two separate customer areas are called swim lanes.

Source: Visible Systems Corporation.

Modeling Tools and Techniques

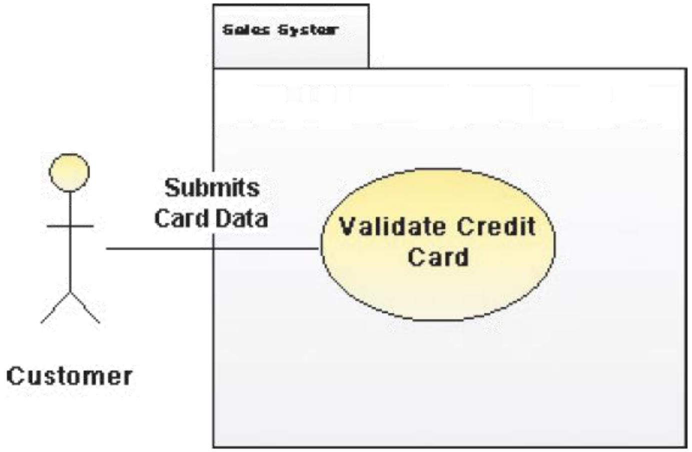


This Visible Analyst DFD shows how books are added and removed in a library system. Source: Visible Systems Corporation.

- **Data Flow Diagrams (DFD)**
  - Show how the system stores, processes, and transforms data
  - Additional levels of information and detail are depicted in other, related DFDs

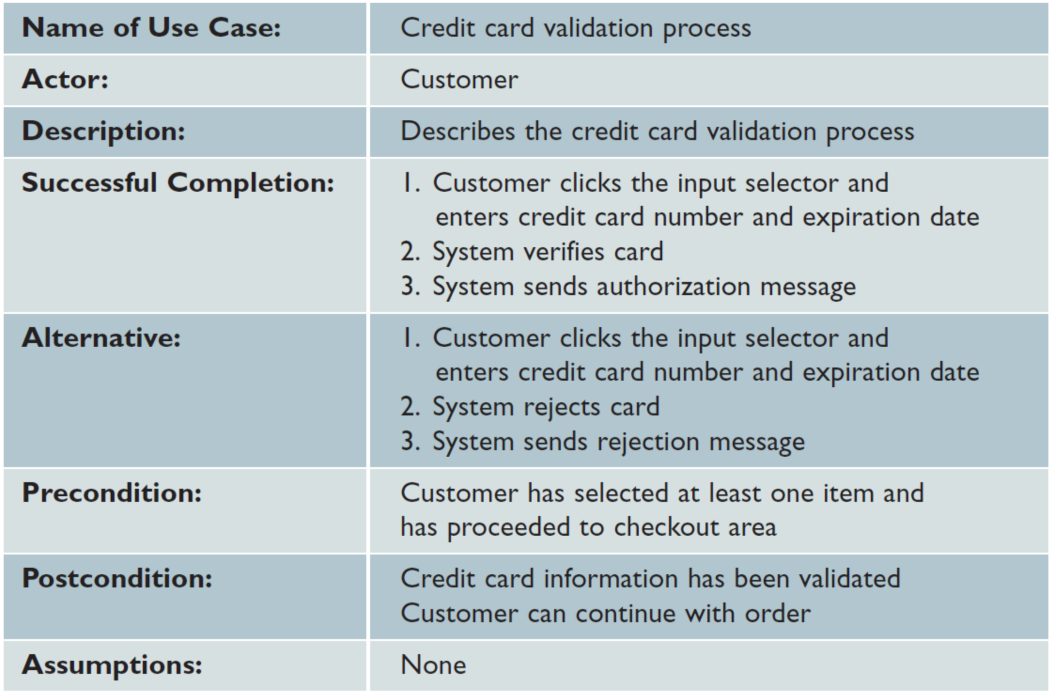
Modeling Tools and Techniques

- **Use Case Diagrams**
  - Represent the interaction between
    users and the system



This Visible Analyst use case diagram shows a sales system, where the actor is a customer and the use case is a credit card validation.

Source: Visible Systems Corporation



Name of Use Case:	Credit card validation process
Actor:	Customer
Description:	Describes the credit card validation process
Successful Completion:	1. Customer clicks the input selector and enters credit card number and expiration date 2. System verifies card 3. System sends authorization message
Alternative:	1. Customer clicks the input selector and enters credit card number and expiration date 2. System rejects card 3. System sends rejection message
Precondition:	Customer has selected at least one item and has proceeded to checkout area
Postcondition:	Credit card information has been validated Customer can continue with order
Assumptions:	None

This table documents the credit card validation use case

UML Use Case Diagram for Modeling Functional Requirements

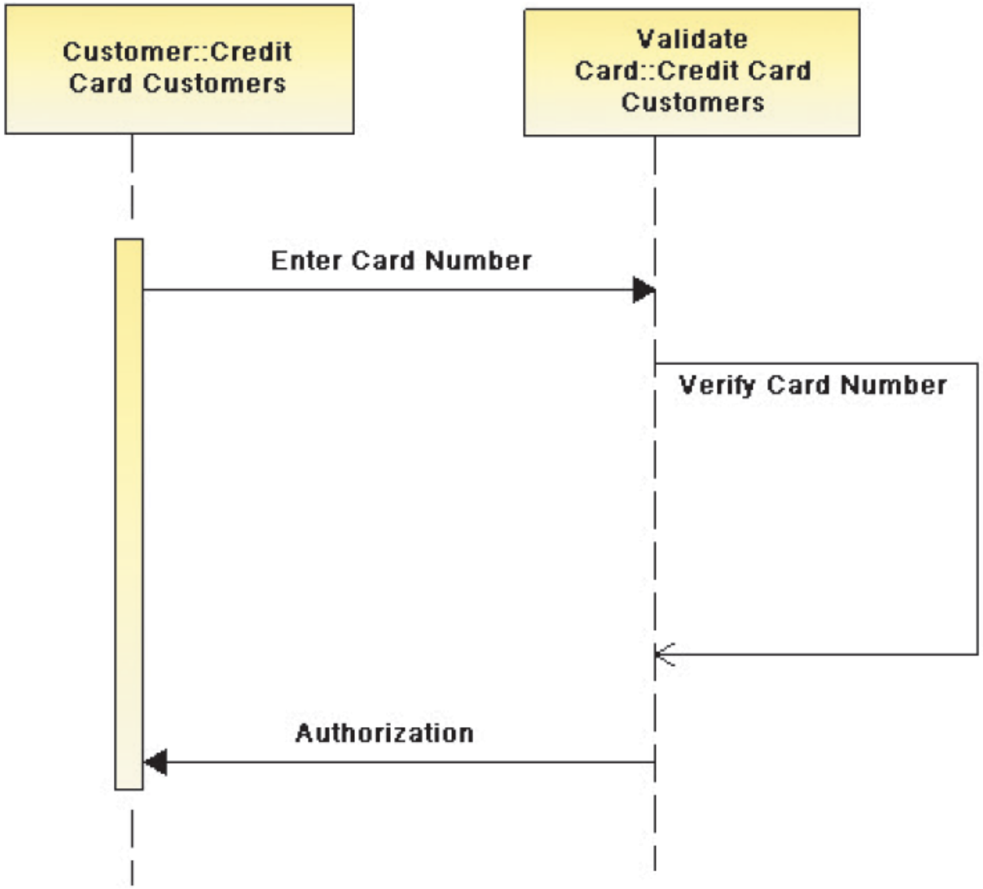
- Most often and traditional way to define functional requirements for the new information system is to model them using UML language and its use case diagrams.

- The requirements must be discussed well with the users.
- One of the advantage is that the symbols used are quite easy to understand even with little experiences in systems development or how systems work.

Adapted from Kniežová, 2013

Modeling Tools and Techniques

- **Sequence Diagrams**
  - Show the timing of interactions between objects as they occur


Customer::Credit Card Customers
Validate Card::Credit Card Customers
Enter Card Number
Verify Card Number
Authorization

This Visible Analyst sequence diagram shows a credit card validation process.

Source: Visible Systems Corporation

System Requirements Checklist

- **Input Examples**
  - The department head must enter overtime hours on a separate screen
  - Student grades must be entered on machine-readable forms prepared by the instructor
  - Each input form must include date, time, product code, customer number, and quantity

- **Process Examples**
  - The student records system must calculate the GPA at the end of each semester
  - The human resources system must interface properly with the existing payroll system
  - The prescription system must automatically generate an insurance claim form

System Requirements Checklist (Cont. 1)

- **Output Examples**
  - The Web site must report online volume statistics every four hours, and hourly during peak periods
  - The contact management system must generate a daily reminder list for all sales reps
  - The purchasing system must provide suppliers with up-to-date specifications

System Requirements Checklist (Cont. 2)

- **Performance Examples**
  - The system must support 25 users online simultaneously
  - Response time must not exceed four seconds
  - The system must be operational seven days a week, 365 days a year

System Requirements Checklist (Cont. 3)

- **Control Examples**
  - The system must provide logon security at the operating system level and at the application level
  - The system must maintain separate levels of security for users and the system administrator
  - All transactions must have audit trails
  - The system must create an error log file that includes the error type, description, and time

Sample Quiz

Assignment 1 Preparation ?

- Meaningful project title

Quiz 1 Preparation ?

Lecture Summary

- At the conclusion of requirements modeling, systems developers should have a clear understanding of business processes and system requirements

- The systems analysis phase includes requirements modeling, data and process modeling, and consideration of development strategies

- Popular team-based approaches include JAD, RAD, and agile methods

Announcement (if any)

Q &A?

Thanks everyone !