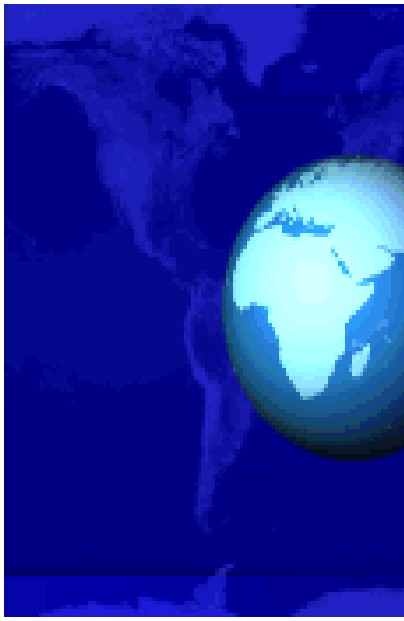


Software Requiremen
Analysis and specifica

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirement Engineering

Requirements describe

What **not** **How**

Produces one large document written in natural
contains a description of what the system will
describing how it will do it.

Crucial process steps

Quality of product  Process that create

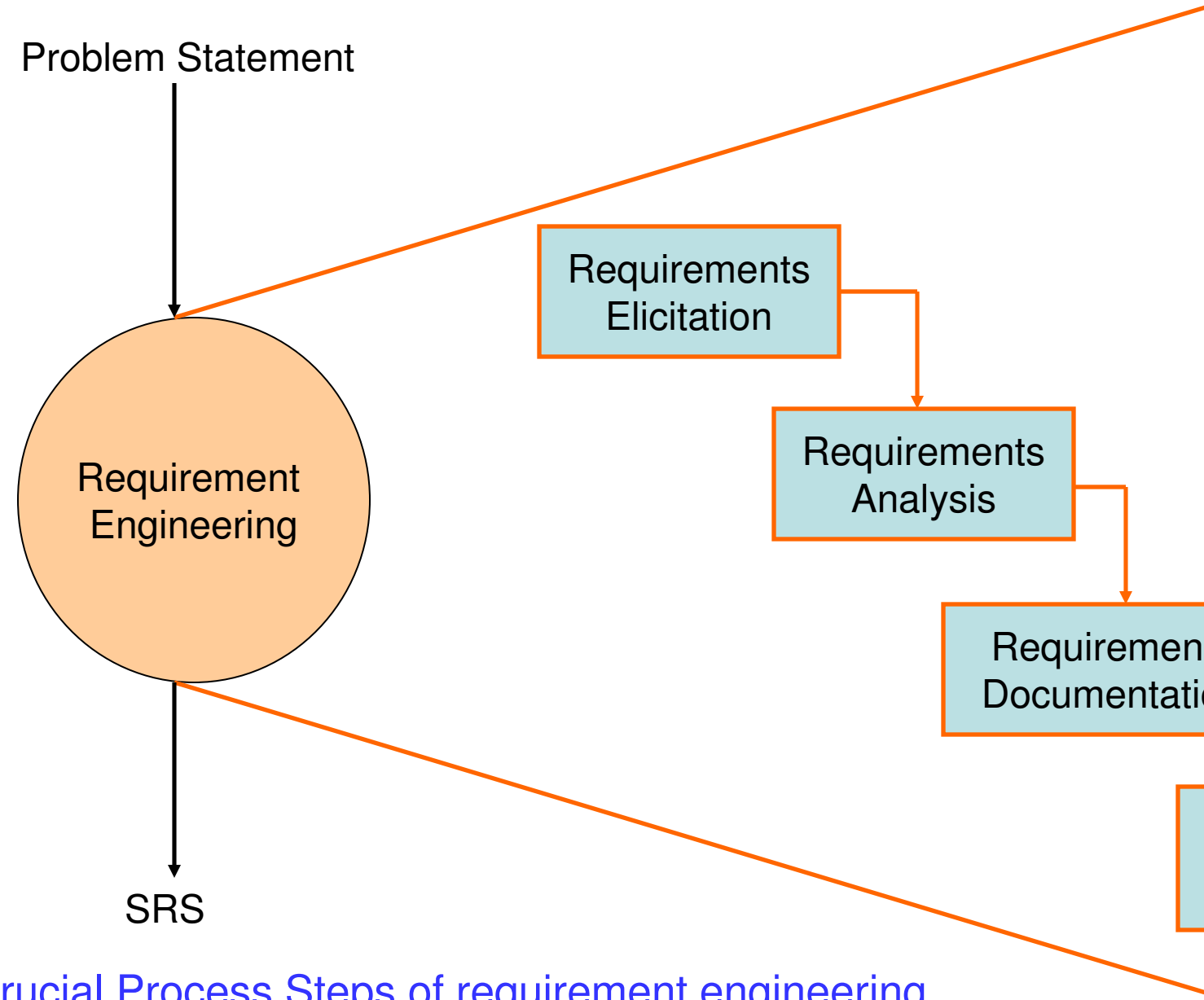
Without well written document

-- Developers do not know what to build

-- Customers do not know what to expect

-- What to validate

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007



Crucial Process Steps of requirement engineering

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirement Engineering

Requirement Engineering is the disciplined app
proven principles, methods, tools, and notations to
proposed system's intended behavior and its
constraints.

SRS may act as a contract between developer and

State of practice

Requirements are difficult to uncover

- Requirements change
- Over reliance on CASE Tools
- Tight project Schedule
- Communication barriers
- Market driven software development
- Lack of resources

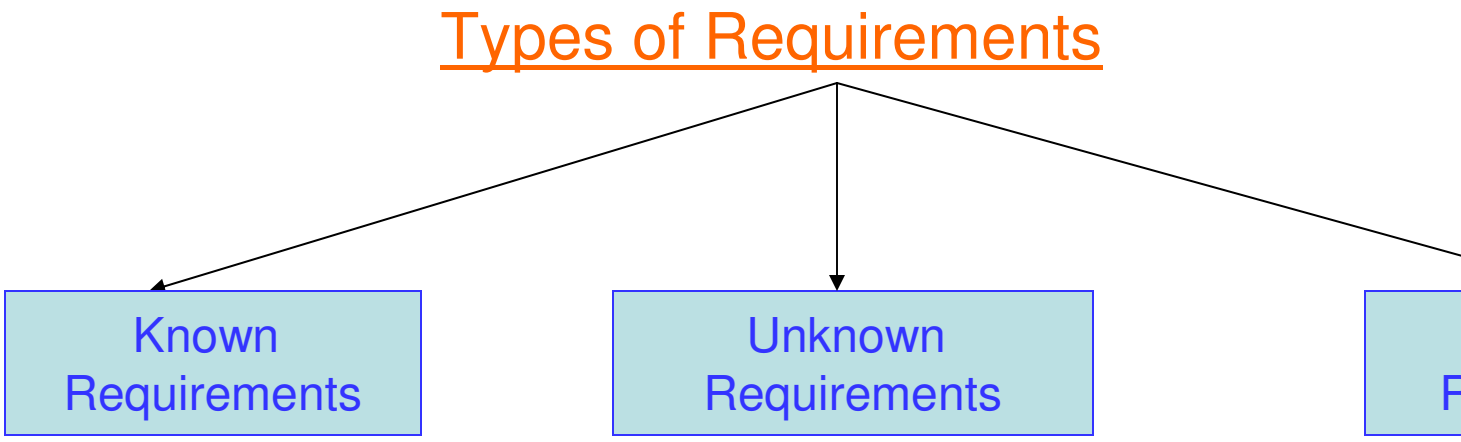
Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Example

A University wish to develop a software systen
student result management of its M.Tech. Progra
problem statement is to be prepared for the
development company. The problem statement
an overview of the existing system and broad exp
from the new software system.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

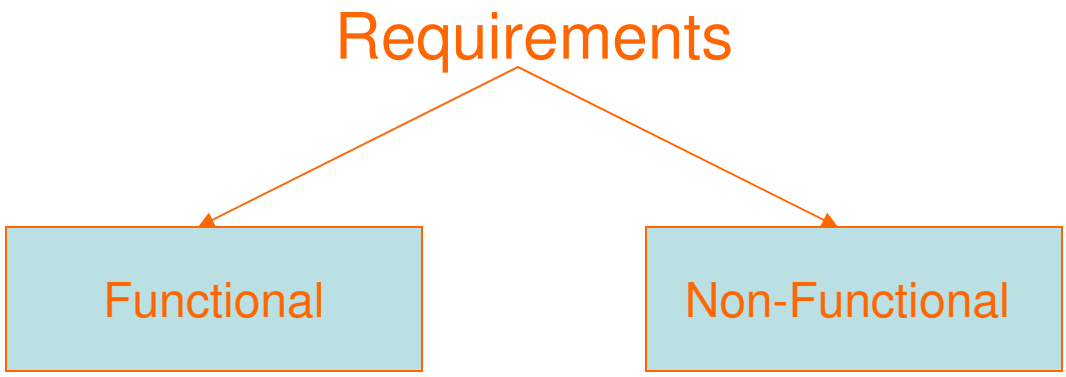
Types of Requirements



Stakeholder: Anyone who should have some direct or ind
influence on the system requirements.

--- User

--- Affected persons



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Types of Requirements

Functional requirements describe what the softwar
do. They are often called product features.

Non Functional requirements are mostly
requirements. That stipulate how well the softwa
what it has to do.

Availability
Reliability
Usability
Flexibility

} For Users

Maintainability
Portability
Testability

} For Developers

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Types of Requirements

User and system requirements

- User requirement are written for the users ar functional and non functional requirement.
- System requirement are derived from user requ
- The user system requirements are the parts o requirement and specification (SRS) document

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Types of Requirements

Interface Specification

- Important for the customers.

TYPES OF INTERFACES

- Procedural interfaces (also called A Programming Interfaces (APIs)).
- Data structures
- Representation of data.

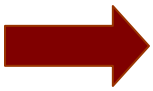
Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Feasibility Study

Is cancellation of a project a bad news?

As per IBM report, “31% projects get cancelled
are completed, 53% over-run their cost estimat
average of 189% & for every 100 projects, there
restarts.

How do we cancel a project with the lea



CONDUCT A FEASIBILTY STUDY

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Feasibility Study

Technical feasibility

- Is it technically feasible to provide direct comm connectivity through space from one location c another location?

- Is it technically feasible to design a pro language using “Sanskrit”?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Feasibility Study

Feasibility depends upon non technical Iss

- Are the project's cost and schedule assumption
- Does the business model realistic?
- Is there any market for the product?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Feasibility Study

Purpose of feasibility study

"evaluation or analysis of the potential imp
proposed project or program."

Focus of feasibility studies

- Is the product concept viable?
- Will it be possible to develop a product that ma
project's vision statement?
- What are the current estimated cost and sched
project?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Feasibility Study

Focus of feasibility studies

- How big is the gap between the original cost & targets & current estimates?
- Is the business model for software justified whe current cost & schedule estimate are considere
- Have the major risks to the project been identifi they be surmounted?
- Is the specifications complete & stable enough support remaining development work?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Feasibility Study

Focus of feasibility studies

- Have users & developers been able to a detailed user interface prototype? If not requirements really stable?
- Is the software development plan complete & to support further development work?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

Perhaps

- Most difficult
- Most critical
- Most error prone
- Most communication intensive

Succeed

effective customer developer partn

Selection of any method

1. It is the only method that we know
2. It is our favorite method for all situations
3. We understand intuitively that the method is effe the present circumstances.

Normally we rely on first two reasons.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

1. Interviews

Both parties have a common goal

--- open ended
--- structured

} Interview

Success of the

Selection of stakeholder

1. Entry level personnel
2. Middle level stakeholder
3. Managers
4. Users of the software (Most important)

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

Types of questions.

- Any problems with existing system
- Any Calculation errors
- Possible reasons for malfunctioning
- No. of Student Enrolled

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

5. Possible benefits
6. Satisfied with current policies
7.How are you maintaining the records of previous st
8. Any requirement of data from other system
9. Any specific problems
10. Any additional functionality
11. Most important goal of the proposed developmen

At the end, we may have wide variety of expectations
proposed software.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

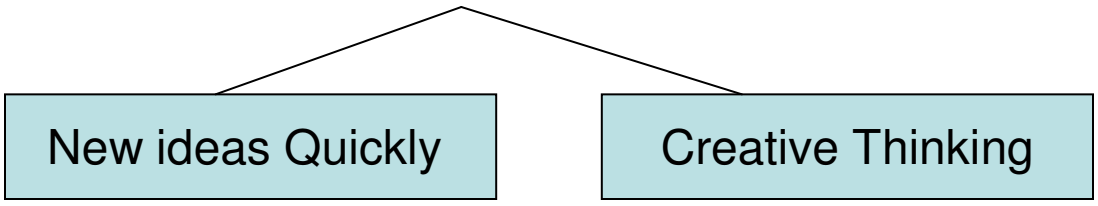
Requirements Elicitation

2. Brainstorming Sessions

It is a group technique

↓

group discussions



Prepare long list of requirements

↓

Categorized
Prioritized
Pruned

*Idea is to generate views ,not to vet them.

Groups

1. Users 2. Middle Level managers 3. Total Stakeho

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

A Facilitator may handle group bias, conflicts ca

-- Facilitator may follow a published agenda

-- Every idea will be documented in a way that every see it.

--A detailed report is prepared.

3. Facilitated Application specification Techniques (F

-- Similar to brainstorming sessions.

-- Team oriented approach

-- Creation of joint team of customers and developer

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

Guidelines

1. Arrange a meeting at a neutral site.
2. Establish rules for participation.
3. Informal agenda to encourage free flow of ideas
4. Appoint a facilitator.
5. Prepare definition mechanism board, worksheet stickier.
6. Participants should not criticize or debate.

FAST session Preparations

Each attendee is asked to make a list of objects th

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

1. Part of environment that surrounds the system.
2. Produced by the system.
3. Used by the system.

A. List of constraints
B. Functions
C. Performance criteria

Activities of FAST session

1. Every participant presents his/her list
2. Combine list for each topic
3. Discussion
4. Consensus list
5. Sub teams for mini specifications
6. Presentations of mini-specifications
7. Validation criteria
8. A sub team to draft specifications

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

4. Quality Function Deployment

-- Incorporate voice of the customer

Technical requirements.

Documented

Prime concern is customer satisfaction

What is important f

-- Normal requirements

-- Expected requirements

-- Exciting requirements

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

Steps

1. Identify stakeholders
2. List out requirements
3. Degree of importance to each requireme

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

5 Points : V. Important

4 Points : Important

3 Points : Not Important but nice to have

2 Points : Not important

1 Points : Unrealistic, required further exploration

Requirement Engineer may categorize like:

(i) It is possible to achieve

(ii) It should be deferred & Why

(iii) It is impossible and should be dropped from consideration

First Category requirements will be implemented a priority assigned with every requirement.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

5. The Use Case Approach

Ivar Jacobson & others introduced Use Case appro
elicitation & modeling.

Use Case – give functional view

The terms

Use Case
Use Case Scenario
Use Case Diagram

} Often Interchanged

But they are differe

Use Cases are structured outline or template f
description of user requirements modeled in a stru
language like English.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

Use case Scenarios are unstructured descriptions
requirements.

Use case diagrams are graphical representations t

may be decomposed into further levels of abstracti

Components of Use Case approach

Actor:

An actor or external agent, lies outside the system
interacts with it in some way.

Actor → Person, machine, information System

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

- Cockburn distinguishes between Prima secondary actors.
- A Primary actor is one having a goal requ assistance of the system.
- A Secondary actor is one from which Syste assistance.

Use Cases

A use case is initiated by a user with a particula mind, and completes successfully when that satisfied.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

* It describes the sequence of interactions betwee and the system necessary to deliver the service satisfies the goal.
* Alternate sequence
* System is treated as black box.

Thus

Use Case captures who (actor) does what (i with the system, for what purpose (goal), witho with system internals.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

*defines all behavior required of the system, bound the scope of the system.

Jacobson & others proposed a template for writing cases as shown below:

1. Introduction

Describe a quick background of the use case.

2.Actors

List the actors that interact and participate in the use cases.

3.Pre Conditions

Pre conditions that need to be satisfied for the use case to perform.

4. Post Conditions

Define the different states in which we expect the system to be in, after the use case executes.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

5. Flow of events

5.1 Basic Flow

List the primary events that will occur when this use case

5.2 Alternative Flows

Any Subsidiary events that can occur in the use case sh
separately listed. List each such event as an alternative f
A use case can have many alternative flows as required.

6.Special Requirements

Business rules should be listed for basic & information flows as sp
requirements in the use case narration .These rules will also be u
for writing test cases. Both success and failures scenarios should
described.

7.Use Case relationships

For Complex systems it is recommended to document the relation
between use cases. Listing the relationships between use cases a
provides a mechanism for traceability

Use Case Template.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Case Guidelines

1. Identify all users
2. Create a user profile for each category of users all roles of the users play that are relevant to the
3. Create a use case for each goal, following the template maintain the same level of abstraction the use case. Steps in higher level use case treated as goals for lower level (i.e. more deta use cases.
4. Structure the use case
5. Review and validate with users.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

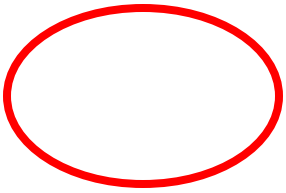
Requirements Elicitation

Use case Diagrams

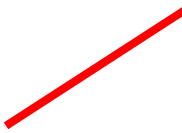
-- represents what happens when actor interacts wi

-- captures functional aspect of the system.

Actor



Use Case



Relationship
actors and us
and/or betwe
use cases.

-- Actors appear outside the rectangle.

--Use cases within rectangle providing functionality

--Relationship association is a solid line between a
cases.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

*Use cases should not be used to capture all the de system.

*Only significant aspects of the required functionalit

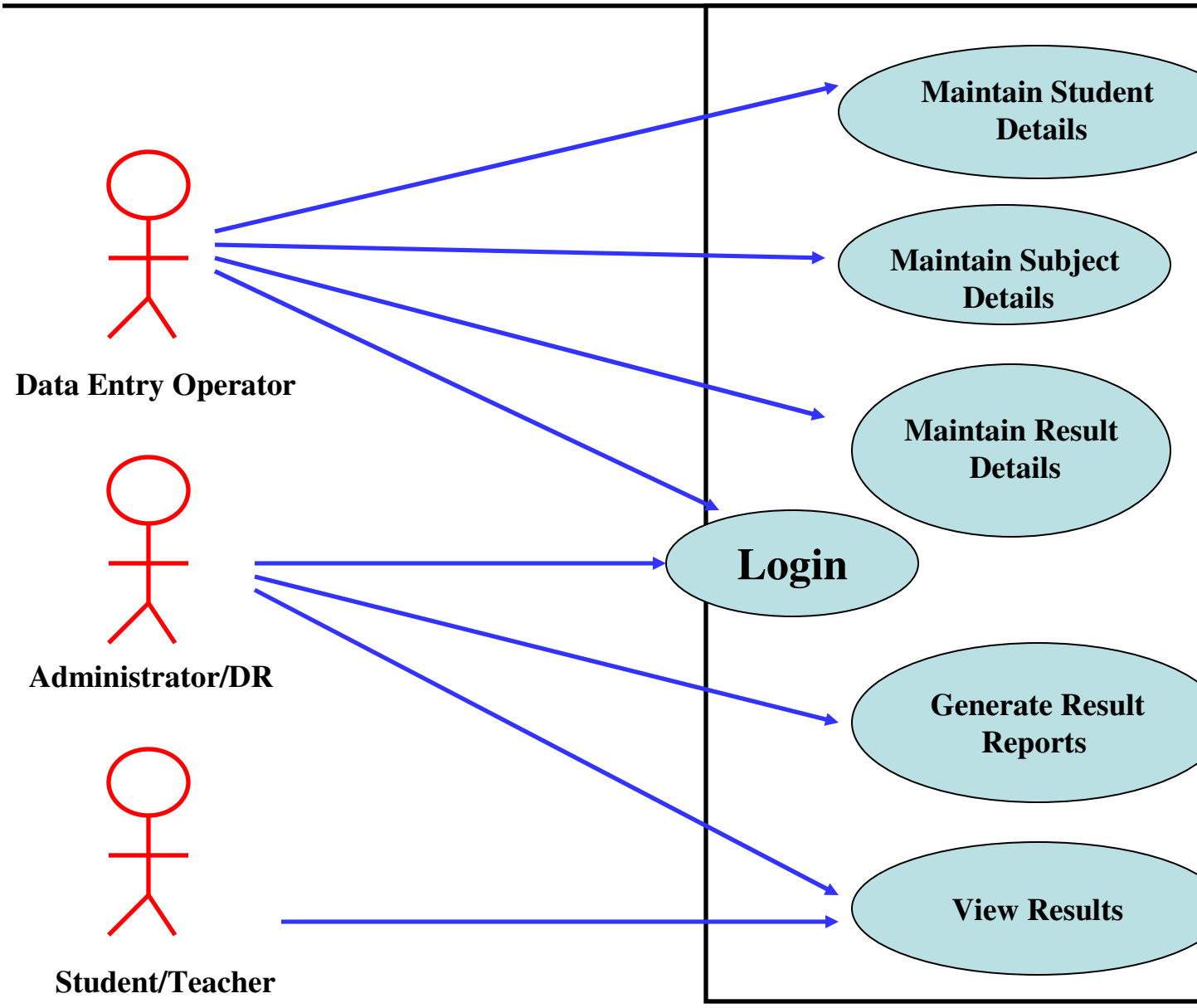
*No design issues

*Use Cases are for “what” the system is , not “how” will be designed

* Free of design characteristics

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use case diagram for Result Management Sys



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation

1. Maintain student Details

   Add/Modify/update students details like name, address.

2. Maintain subject Details

   Add/Modify/Update Subject information semester wise

3. Maintain Result Details

   Include entry of marks and assignment of credit points paper.

4. Login

   Use to Provide way to enter through user id & password

5. Generate Result Report

   Use to print various reports

6. View Result
   (i) According to course code
   (ii) According to Enrollment number/roll number

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation (Use Case)

Login

1.1 Introduction : This use case describes how a u
the Result Management System.

1.2 Actors : (i) Data Entry Operator
(ii) Administrator/Deputy Regi

1.3 Pre Conditions : None

1.4 Post Conditions : If the use case is successful
logged into the system. If not, the system state is u

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Elicitation (Use Case)

1.5 Basic Flow : This use case starts when the acto
to login to the Result Management system.

(i) System requests that the actor enter his/her name
password.

(ii) The actor enters his/her name & password.

(iii) System validates name & password, and if finds
allow the actor to logs into the system.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

1.6 Alternate Flows

1.6.1 Invalid name & password

If in the basic flow, the actor enters an inva
and/or password, the system displays an error mess
actor can choose to either return to the beginning of
flow or cancel the login, at that point, the use case e

1.7 Special Requirements:

None

1.8 Use case Relationships:

None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

2.Maintain student details

2.1 Introduction : Allow DEO to maintain stu
This includes adding, changing and deleting studer
information

2.2 Actors : DEO

2.3 Pre-Conditions: DEO must be logged or
system before this use case begins.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

2.4 Post-conditions : If use case is succes
information is added/updated/deleted from t
Otherwise, the system state is unchanged.

2.5 Basic Flow : Starts when DEO
add/modify/update/delete Student information.

(i) The system requests the DEO to specify t
he/she would like to perform (Add/update/delet

(ii) One of the sub flow will execute after
information.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

- If DEO selects "Add a student", "Add a student" sub
  be executed.

- If DEO selects "update a student", "update a student
  will be executed.

- If DEO selects "delete a student", "delete a student"
  will be executed.

2.5.1 Add a student

(i) The system requests the DEO to enter:
- Name
- Address
- Roll No
- Phone No
- Date of admission

(ii) System generates unique id

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

2.5.2 Update a student

(i) System requires the DEO to enter student-id.

(ii) DEO enters the student_id. The system ret
display the student information.

(iii) DEO makes the desired changes to th
information.

(iv) After changes, the system updates the student
changed information.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

2.5.3 Delete a student

(i) The system requests the DEO to specify the stud

(ii) DEO enters the student-id. The system retri displays the student information.

(iii) The system prompts the DEO to confirm the d the student.

(iv) The DEO confirms the deletion.

(v) The system marks the student record for deletion

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

2.6 Alternative flows

2.6.1 Student not found

If in the update a student or delete a student a student with specified_id does not exist, t displays an error message .The DEO ma different id or cancel the operation. At this case ends.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

2.6.2 Update Cancelled

If in the update a student sub-flow, the
operator decides not to update the student ir
the update is cancelled and the basic flow is r
the begin.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

2.6.3 Delete cancelled

If in the delete a student sub flows, DEO dec
delete student record ,the delete is cancelle
basic flow is re-started at the beginning.

2.7 Special requirements

None

2.8 Use case relationships

None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

3. Maintain Subject Details

3.1 Introduction

The DEO to maintain subject information. This
adding, changing, deleting subject information
system

3.2 Actors : DEO

3.3 Preconditions : DEO must be logged onto the system before the use case beg

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

3.4 Post conditions :

If the use case is successful, the subject inf
is added, updated, or deleted from the
otherwise the system state is unchanged.

3.5 Basic flows :

The use case starts when DEO wishes to add,
and/or delete subject information from the system

(i) The system requests DEO to specify the functio
would like to perform i.e.

- Add a subject
- Update a subject
- Delete a subject.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

(ii) Once the DEO provides the required information, one flows is executed.

- If DEO selected “Add a subject” the “Add-a subje is executed.
- If DEO selected “Update-a subject” the “update-a sub flow is executed
- If DEO selected “Delete- a- subject”, the “Delete- sub flow is executed.

3.5.1 Add a Subject

(i) The System requests the DEO to enter th subject information. This includes :

* Name of the subject

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

* Subject Code
* Semester
* Credit points

(ii) Once DEO provides the requested inf
the system generates and assigns a unique subject
subject. The subject is added to the system.

(iii) The system provides the DEO with ne
subject-id.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

3.5.2 Update a Subject

(i) The system requests the DEO to subject_id.

(ii) DEO enters the subject_id. The system r and displays the subject information.

(iii) DEO makes the changes.

(iv) Record is updated.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

3.5.3 Delete a Subject

(i) Entry of subject_id.

(ii) After this, system retrieves & displays subjec information.

* System prompts the DEO to confirm the de

* DEO verifies the deletion.

* The system marks the subject record for d

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

3.6 Alternative Flow

3.6.1 Subject not found

If in any sub flows, subject-id not found, error m
displayed. The DEO may enter a different id or c
case ends here.

3.6.2 Update Cancelled

If in the update a subject sub-flow, the data entry
decides not to update the subject information, the
cancelled and the basic flow is restarted at the beg

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

3.6.3 Delete Cancellation

If in delete-a-subject sub flow, the DEO decide
delete subject, the delete is cancelled, and the ba
is restarted from the beginning.

3.7 Special Requirements:

None

3.8 Use Case-relationships

None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

4. Maintain Result Details

4.1 Introduction

This use case allows the DEO to maintain s
marks information of each student. This includes a
and/or deleting subject and marks information fro
system.

4.2 Actor

DEO

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

4.3 Pre Conditions

DEO must be logged onto the systen

4.4 Post Conditions

If use case is successful ,marks infor
added or deleted from the system. O
the system state is unchanged.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

4.5 Basic Flow

This use case starts, when the DEO wishes to update and/or delete marks from the system.

(i) DEO to specify the function

(ii) Once DEO provides the information one of t subflow is executed.

* If DEO selected “Add Marks “, the Add mark subflow is executed.
* If DEO selected “Update Marks”, the update subflow is executed.
* If DEO selected “Delete Marks”, the delete subflow is executed.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

4.5.1 Add Marks Records

Add marks information .This includes:

a. Selecting a subject code.

b. Selecting the student enrollment number.

c. Entering internal/external marks for that subjec
enrollment number.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

(ii) If DEO tries to enter marks for the same combina
subject and enrollment number,the system gives a n
that the marks have already been entered.

(iii) Each record is assigned a unique result_id.

4.5.2 Delete Marks records

1. DEO makes the following entries:

   a. Selecting subject for which marks ha
      deleted.

   b. Selecting student enrollment numbe

   c. Displays the record with id number.

   d. Verify the deletion.

   e. Delete the record.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

4.5.2 Update Marks records

1. The System requests DEO to enter the record_id.

2. DEO enters record_id. The system retrie displays the information.

3. DEO makes changes.

4. Record is updated.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

4.5.3 Compute Result

(i) Once the marks are entered, result is c
for each student.

(ii) If a student has scored more than 50%
subject, the associated credit points are allotted to
student.

(iii) The result is displayed with subject-cod
& credit points.

4.6 Alternative Flow

4.6.1 Record not found

If in update or delete marks sub flows
with specified id number do not exist, the system c
an error message. DEO can enter another id or ca
operation.

4.6.2 Delete Cancelled

If in Delete Marks, DEO decides not to delete the delete is cancelled and basic flow is re-started a beginning.

4.7 Special Requirements

None

4.8 Use case relationships

None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

5 View/Display result

5.1 Introduction

This use case allows the student/Teacher o
to view the result. The result can be viewed on t
of course code and/or enrollment number.

5.2 Actors

Administrator/DR, Teacher/Student

5.3 Pre Conditions

None

5.4 Post Conditions

If use case is successful, the marks informa
displayed by the system. Otherwise, state is un

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

5.5 Basic Flow

Use case begins when student, teacher or any o
person wish to view the result.

Two ways

-- Enrollment no.

-- Course code

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

(ii) After selection, one of the sub flow is execu

Course code → Sub flow is execut

Enrollment no. → Sub flow is execut

5.5.1 View result enrollment number wise

(i) User to enter enrollment number

(ii) System retrieves the marks of all subjects credit points

(iii) Result is displayed.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

5.6 Alternative Flow

5.6.1 Record not found

Error message should be displaye

5.7 Special Requirements

None

5.8 Use Case relationships

None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

6. Generate Report

6.1 Introduction

This use case allows the DR to generate re
reports. Options are

a. Course code wise
b. Semester wise
c. Enrollment Number wise

6.2 Actors

DR

6.3 Pre-Conditions

DR must logged on to the system

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

6.4 Post conditions

If use case is successful, desired report is generated. Otherwise, the system state is unchanged.

6.5 Basic Flow

The use case starts, when DR wish to gener reports.

(i) DR selects option.

(ii) System retrieves the information displays

(iii) DR takes printed reports.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

6.6 Alternative Flows

6.6.1 Record not found

If not found, system generates ap
message. The DR can select another option or c
operation. At this point, the use case ends.

6.7 Special Requirements

None

6.8 Use case relationships

None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

7. Maintain User Accounts

7.1 Introduction

This use case allows the administrator to
user account. This includes adding, chan
deleting user account information from the sys

7.2 Actors

Administrator

7.3 Pre-Conditions

The administrator must be logged o
system before the use case begins.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

7.4 Post-Conditions

If the use case was successful, the user account is added, updated, or deleted from the system. Oth system state is unchanged.

7.5 Basic Flow

This use case starts when the Administrator wishe change, and/or delete use account information from the s

(i) The system requests that the Administrator sp function he/she would like to perform (either Ad Account, Update a User Account, or Delete Account).

(ii) Once the Administrator provides the requested inf one of the sub-flows is executed

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

* If the Administrator selected “Add a User Ac
Add a User Account sub flow is executed.

* If the Administrator selected “Update a User Ac
Update a User Account sub-flow is executed.

* If the Administrator selected “Delete a User Ac
Delete a User Account sub-flow is executed.2

7.5.1 Add a User Account

1. The system requests that the Administrator ente
information. This includes:

(*a*) User Name

(*b*) User ID-should be unique for each user acc

(*c*) Password

(*d*) Role

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

2. Once the Administrator provides the requested infor
   user account information is added.

7.5.2 Update a User Account

1. The system requests that the Administrator enters th
2. The Administrator enters the User ID. The system re
   displays the user account information.
3. The Administrator makes the desired changes t
   account information. This includes any of the
   specified in the **Add a User Account** sub-flow.
4. Once the Administrator updates the necessary infor
   system updates the user account record with th
   information.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

7.5.3 Delete a User Account

1. The system requests that the Administra the User ID.
2. The Administrator enters the User ID. Th retrieves and displays the user account inf
3. The system prompts the Administrator t the deletion of the user account.
4. The Administrator confirms the deletion.
5. The system deletes the user account recor

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

7.6 Alternative Flows

7.6.1 User Not Found

If in the **Update a User Account** or **Delet Account** sub-flows, a user account with the User ID does not exist, the system displays message. The Administrator can then different User ID or cancel the operation, at whic the use case ends.

7.6.2 Update Cancelled

If in the **Update a User Account** sub Administrator decides not to update the use information, the update is cancelled and the **Bas** is re-started at the beginning.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

7.6.3 Delete Cancelled

If in the **Delete a User Account** sub
Administrator decides not to delete the us
information, the delete is cancelled and
Flow is re-started at the beginning.

7.7 Special Requirements

None

7.8 Use case relationships

None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

8. Reset System

8.1 Introduction

This use case allows the allows the adminis
reset the system by deleting all existing informa
the system .

8.2 Actors

Administrator

8.3 Pre-Conditions

The administrator must be logged on to the
before the use case begins.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

8.4 Post-Conditions

If the use case was successful, all th
information is deleted from the backend da
the system. Otherwise, the system state is un

8.5 Basic Flow

This use case starts when the Administrator w
reset the system.

i. The system requests the Administrator to
he/she wants to delete all the existing i
from the system.

ii. Once the Administrator provides confirm
system deletes all the existing informatio
backend database and displays an a
message.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Use Cases

8.6 Alternative Flows

8.6.1 Reset Cancelled

If in the Basic Flow, the Administrator
not to delete the entire existing information, the res
cancelled and the use case ends.

8.7 Special Requirements

None

8.8 Use case relationships

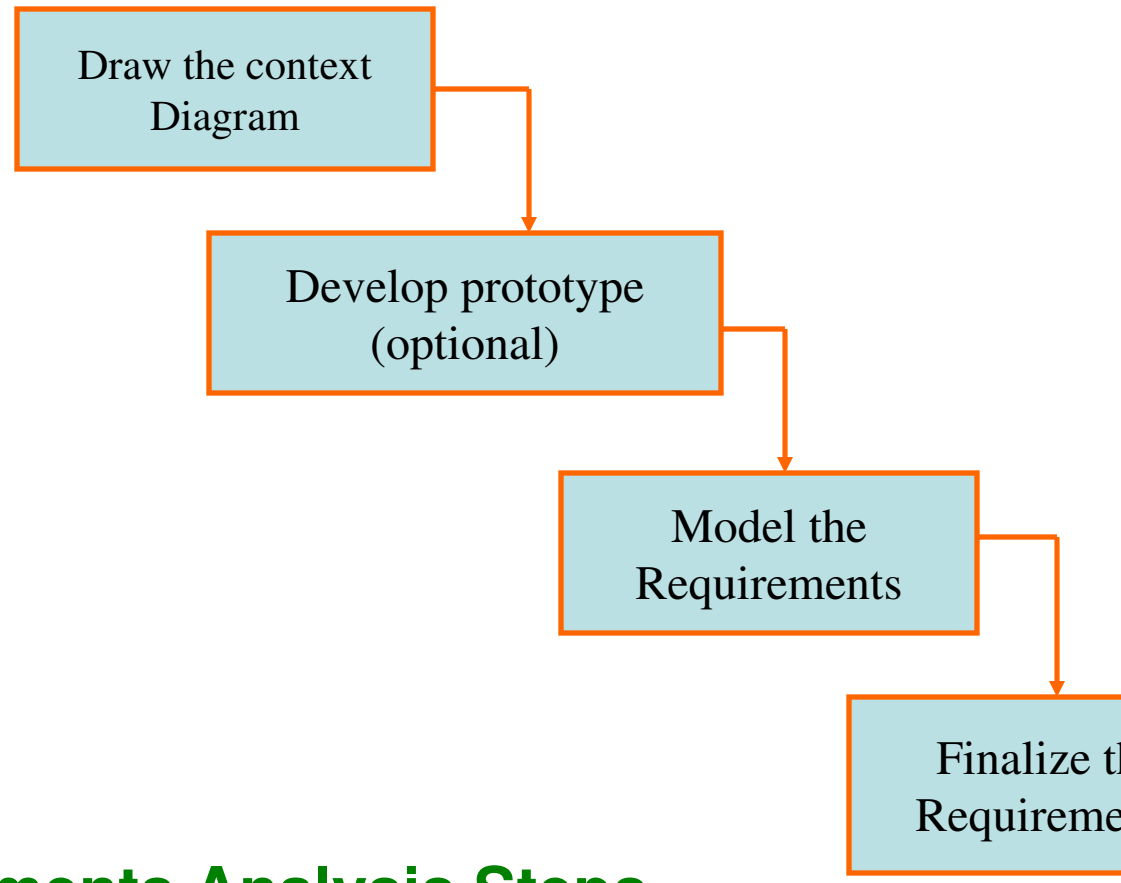
None

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Analysis

We analyze, refine and scrutinize requirements t
consistent & unambiguous requirements.

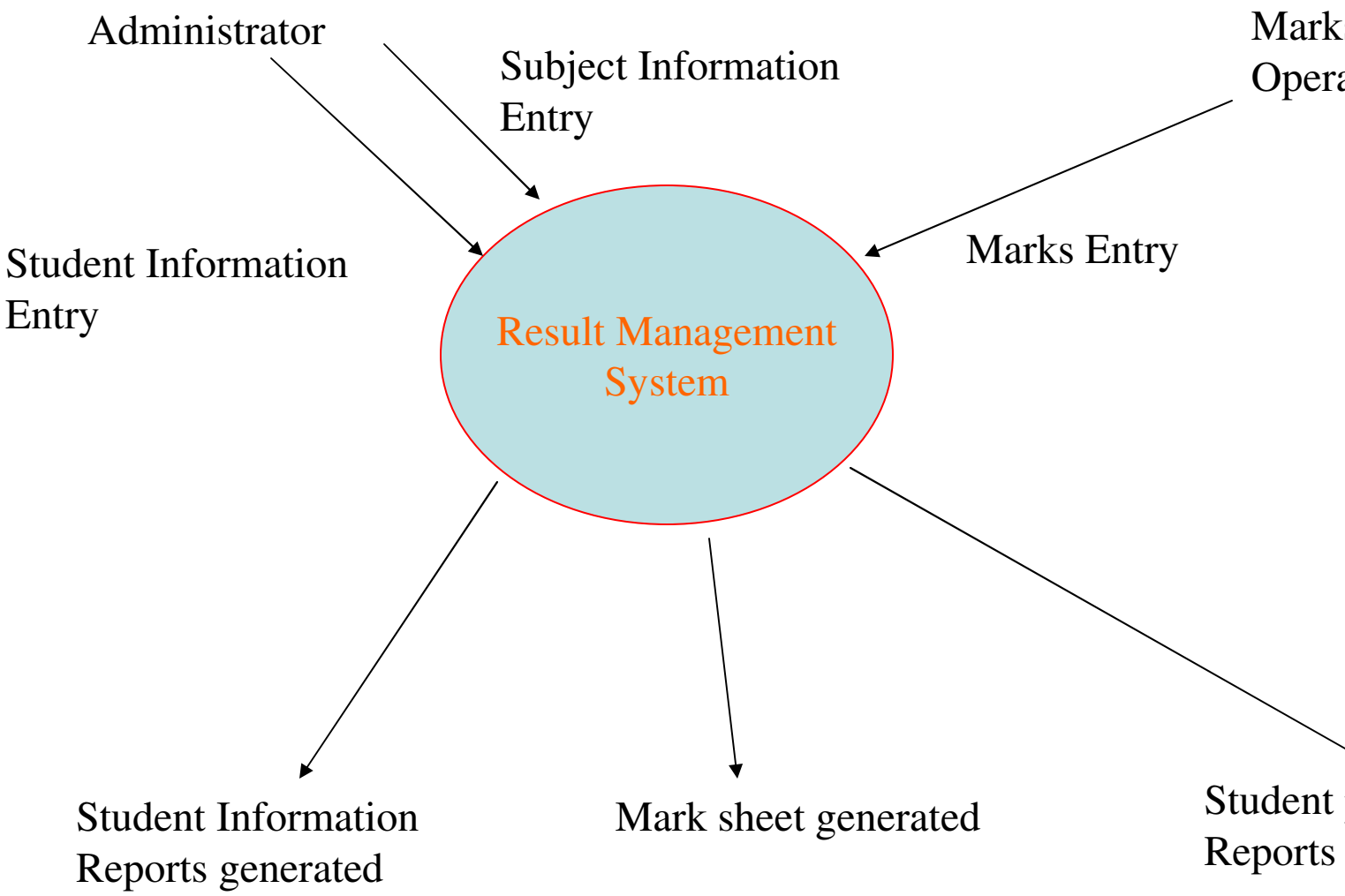
Steps



Requirements Analysis Steps

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Analysis



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Data Flow Diagrams

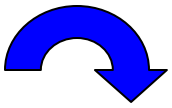
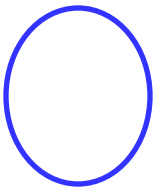
DFD show the flow of data through the system.

--All names should be unique

-- It is not a flow chart

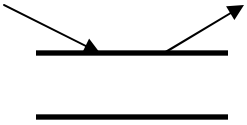
-- Suppress logical decisions

-- Defer error conditions & handling until the
the analysis

Symbol	Name	Function
	Data Flow	Connect process
	Process	Perform some transform input data to yield output

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Analysis

Symbol	Name	Functio
	Source or sink	A source of syst sink of system o
	Data Store	A repository of d arrowhead indic input and net ou to store

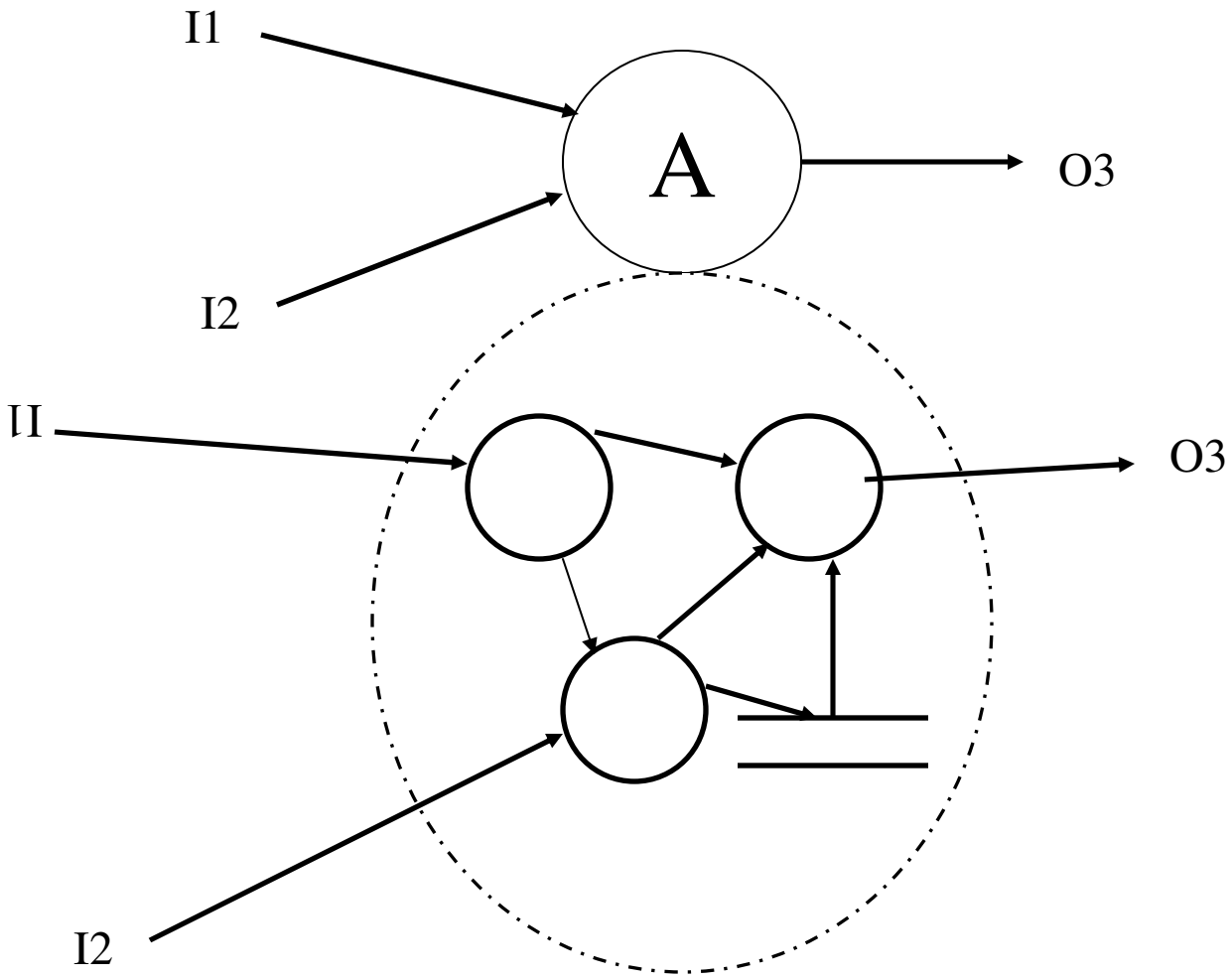
Leveling

DFD represent a system or software at any level of abstraction.

A level 0 DFD is called fundamental system model model represents entire software element as a sing with input and output data indicating by incoming & arrows.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Analysis



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Data Dictionaries

DFD ⟶ DD

Data Dictionaries are simply repositories to store information about all data items defined in DFD.

Includes :

- Name of data item
- Aliases (other names for items)
- Description/Purpose
- Related data items
- Range of values
- Data flows
- Data structure definition

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Data Dictionaries

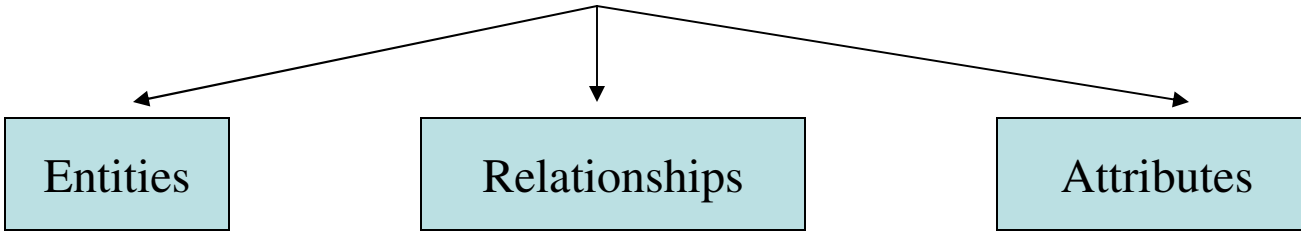
Notation	Meaning
x= a+b	x consists of data element a & b
x={a/b}	x consists of either a or b
x=(a)	x consists of an optional data elemen
x= y{a}	x consists of y or more occurrences
x={a}z	x consists of z or fewer occurrences o
x=y{a}z	x consists of between y & z occurren

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams

Entity-Relationship Diagrams

It is a detailed logical representation of data for an organization and uses three main constructs.



Entities

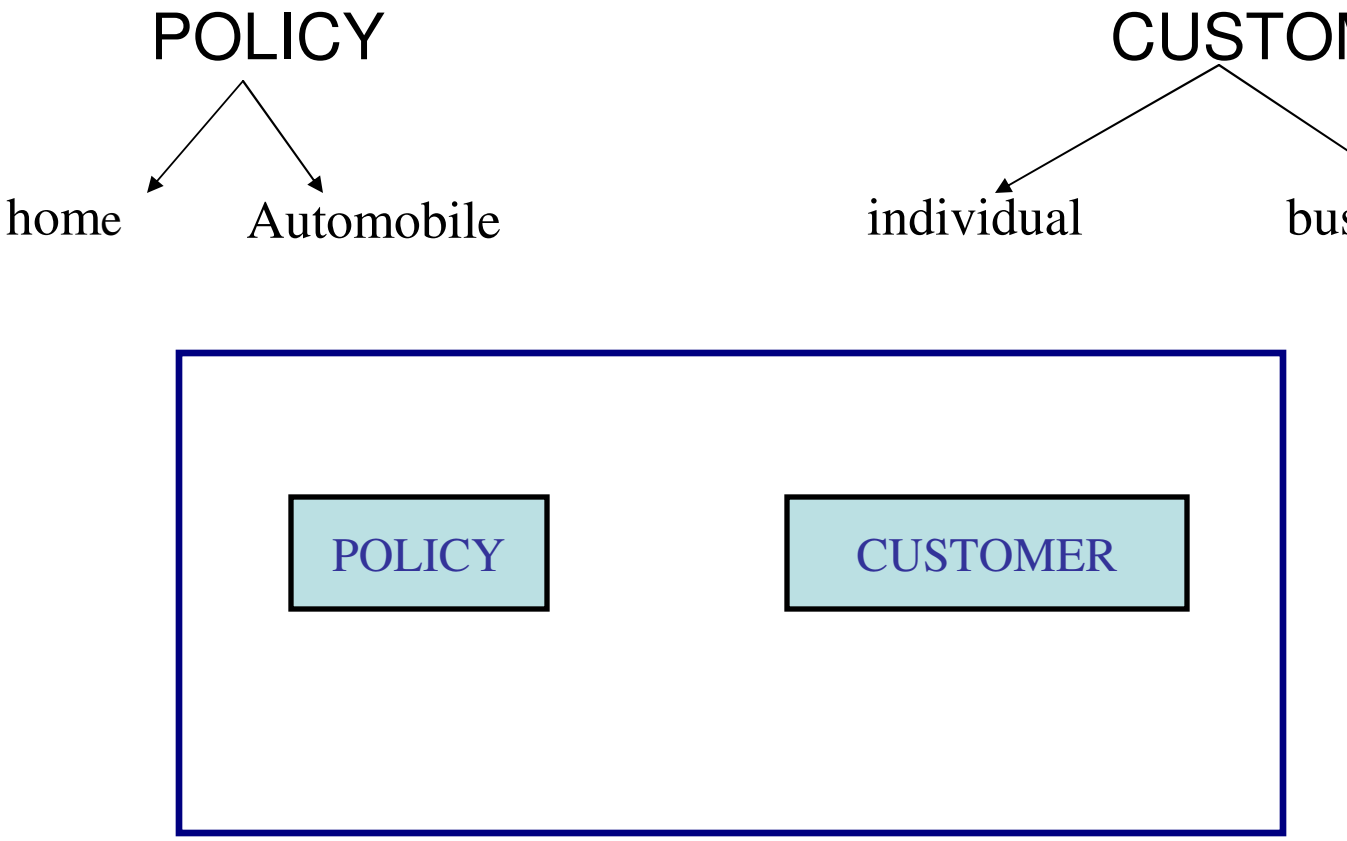
Fundamental thing about which data maintained. Each entity has its own identity.

Entity Type is the description of all entities to common definition and common relationships and apply.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams

Consider an insurance company that offers both l
and automobile insurance policies .These policies
offered to individuals and businesses.



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams

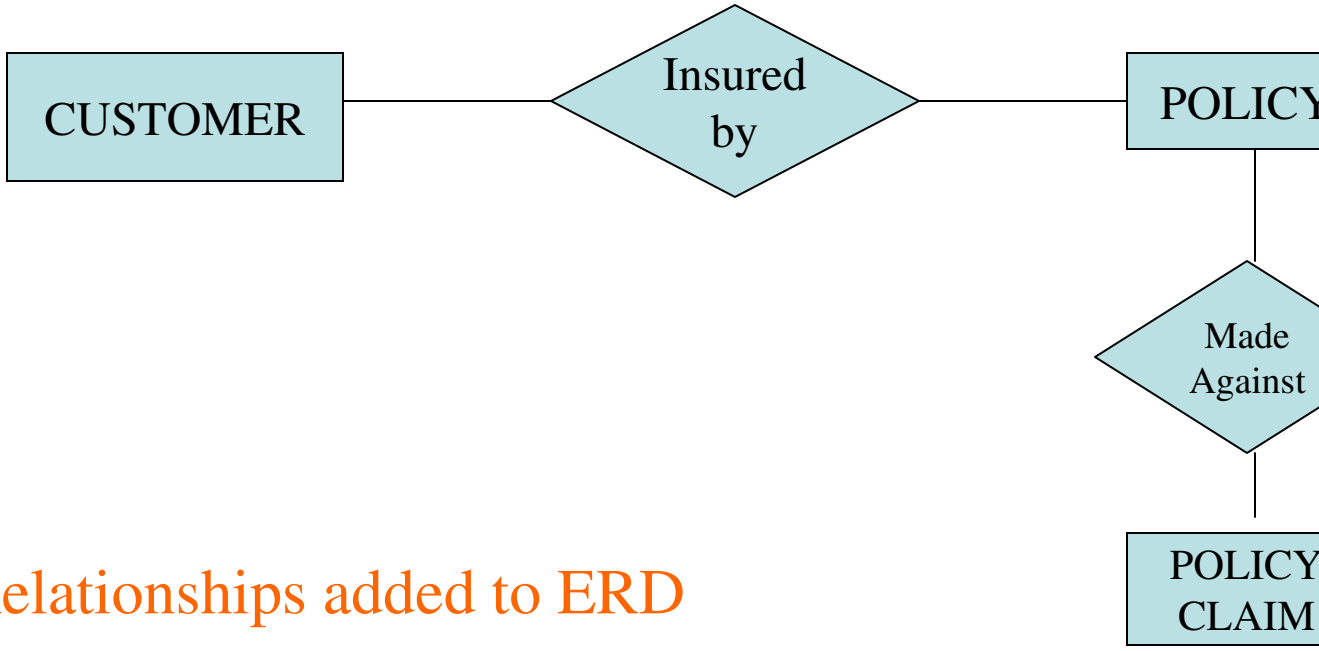
Relationships

A relationship is a reason for associating two entity types.

Binary relationships ⟶ involve two entity types

A CUSTOMER is insured by a POLICY. A POLICY CLAIM is against a POLICY.

Relationships are represented by diamond notation in a ER d

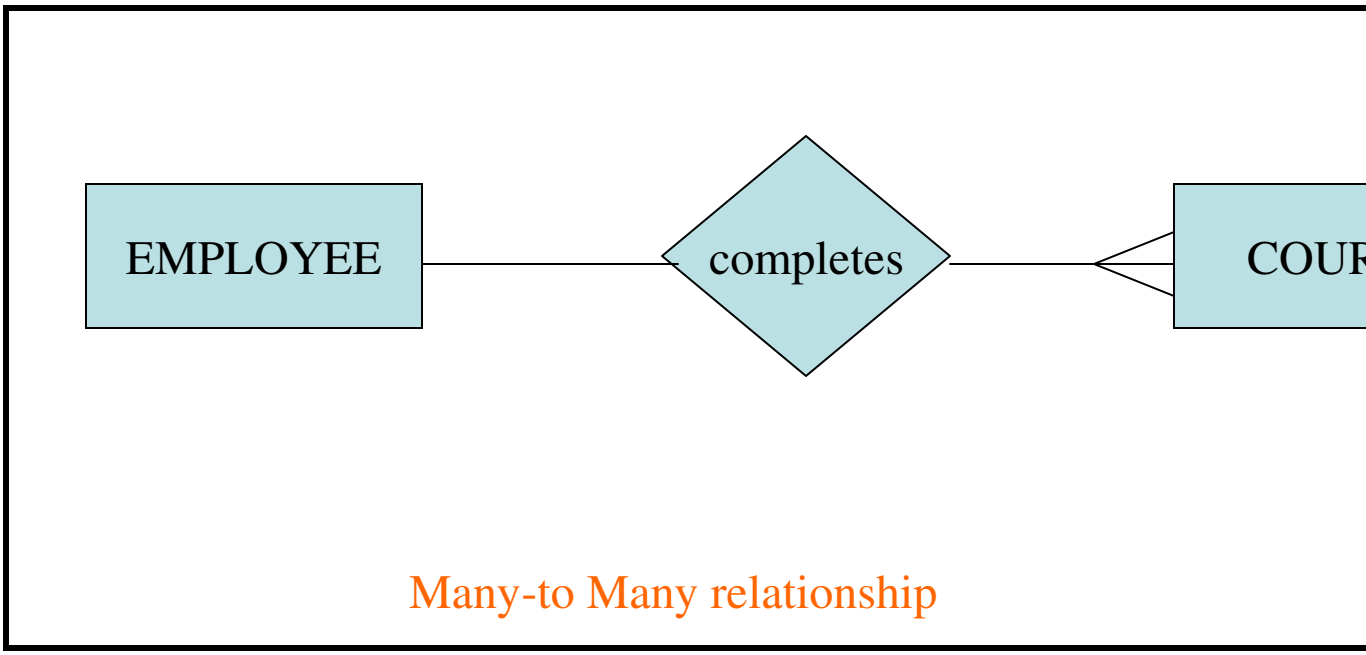


Relationships added to ERD

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams

A training department is interested in tracking which courses each of its employee has completed.



Many-to Many relationship

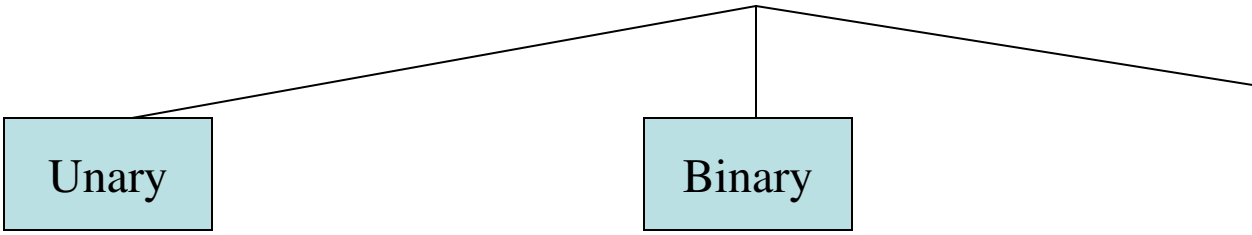
Each employee may complete more than one cour each course may be completed by more tha employee.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

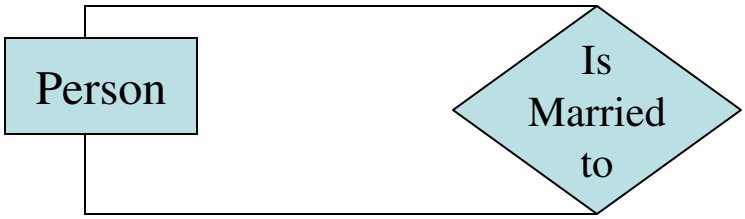
Entity-Relationship Diagrams

Degree of relationship

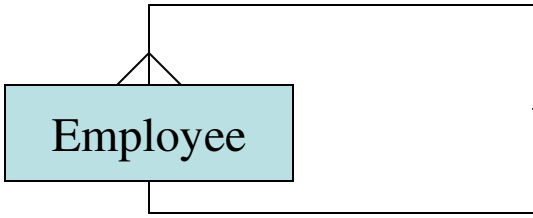
It is the number of entity types that participates in that relationshi



Unary relationship



One to One

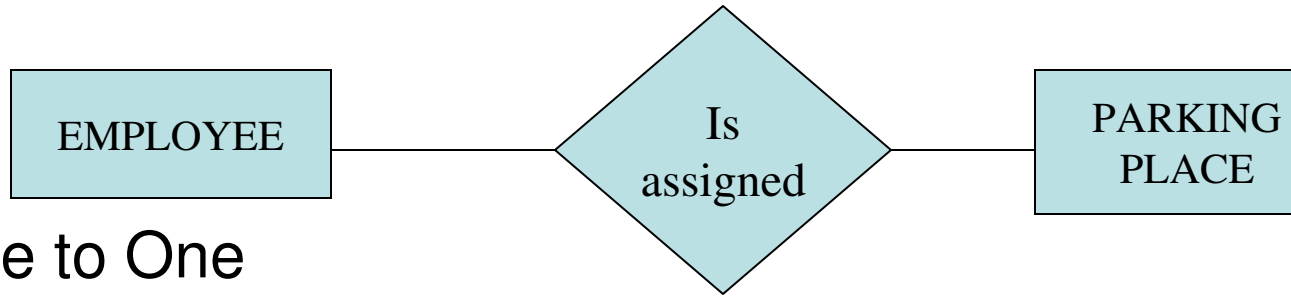


One to ma

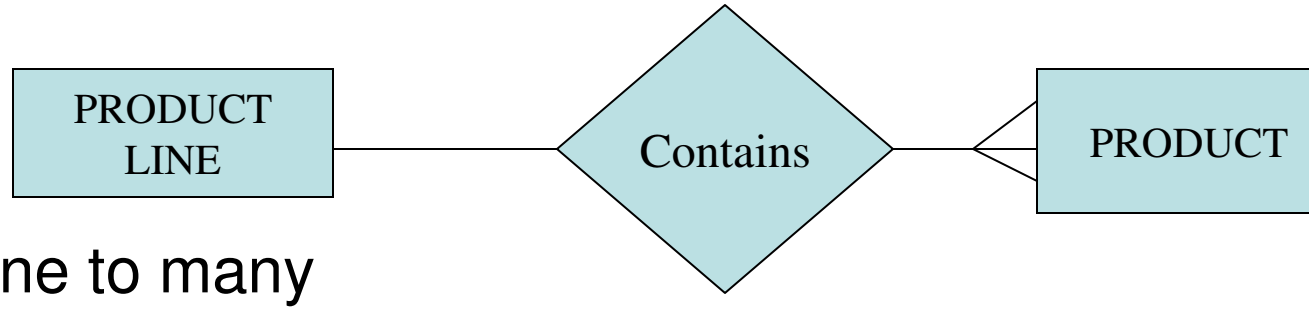
Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams

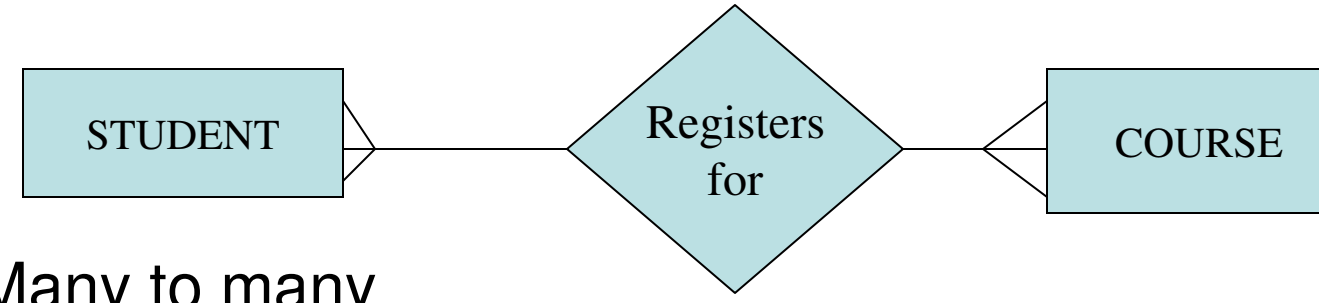
Binary Relationship



One to One



One to many

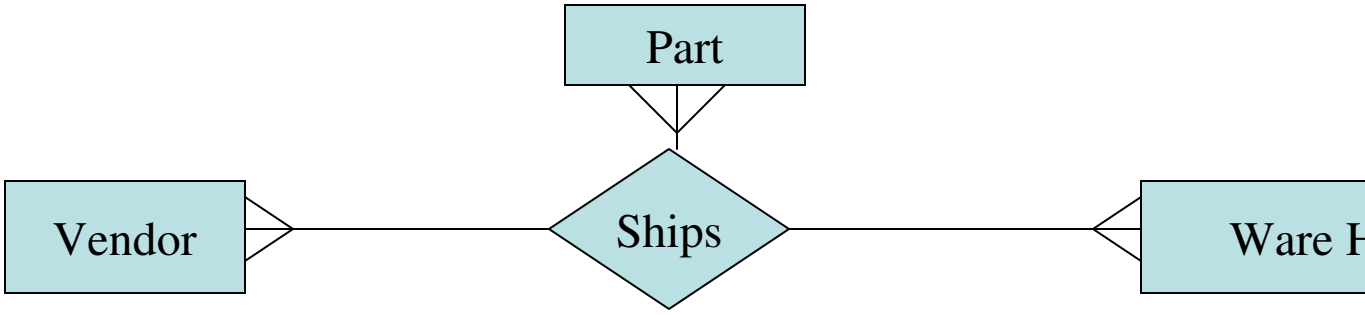


Many to many

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

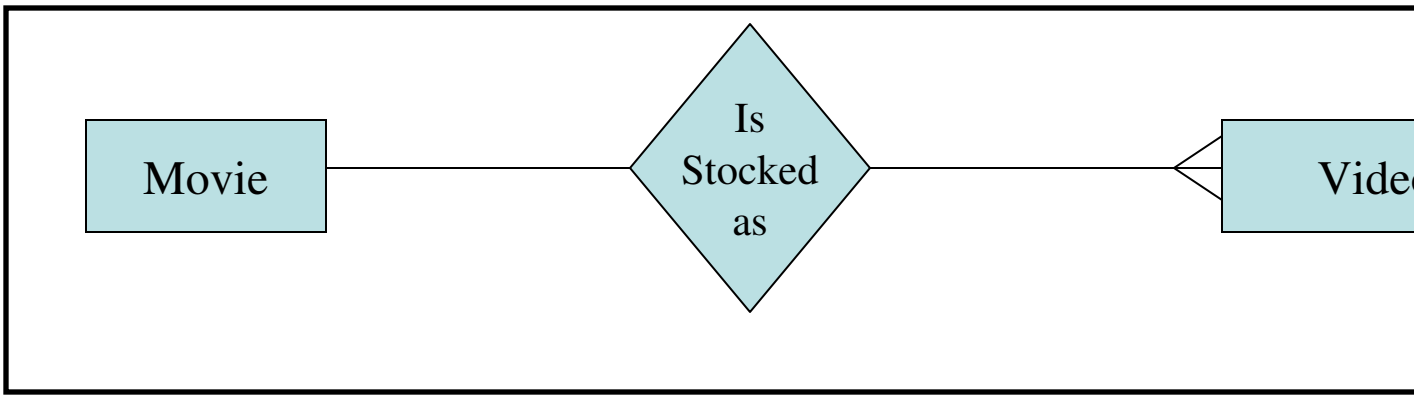
Entity-Relationship Diagrams

Ternary relationship



Cardinalities and optionality

Two entity types A,B, connected by a relationship.
The cardinality of a relationship is the number of instances of e
can be associated with each instance of entity A

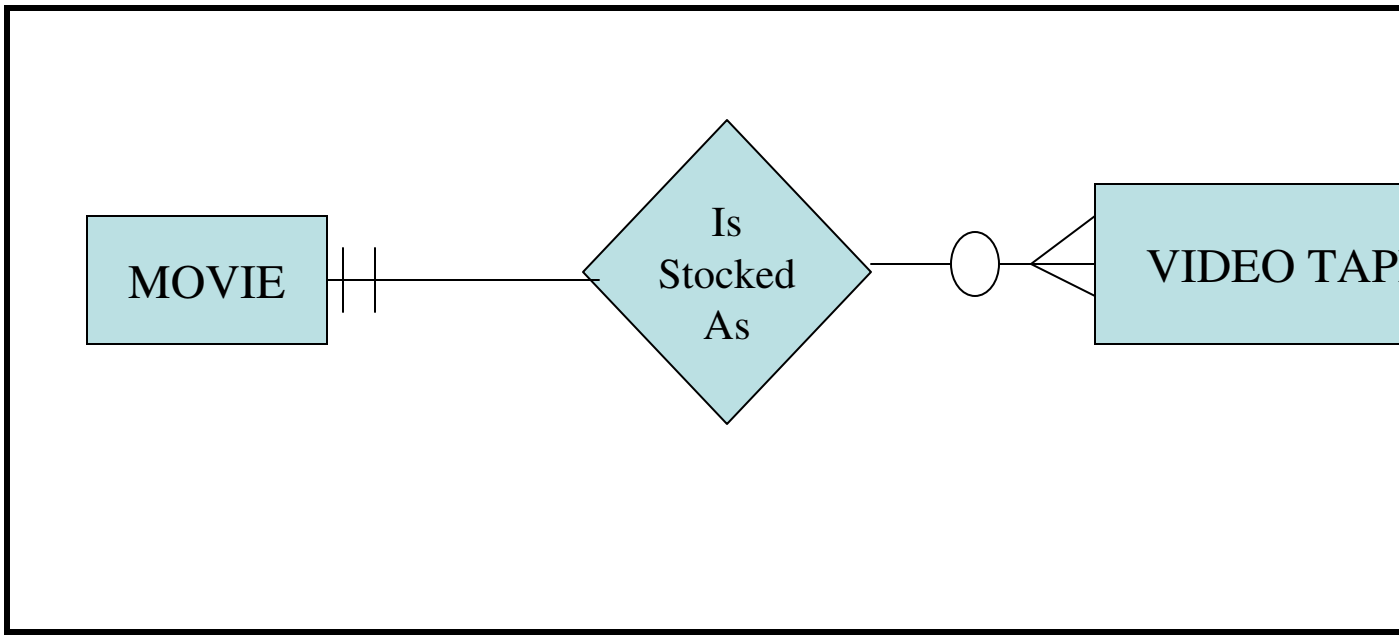


Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams

Minimum cardinality is the minimum number of ins
entity B that may be associated with each instance
A.

Minimum no. of tapes available for a movie is zer
VIDEO TAPE is an optional participant in the is-st
relationship.



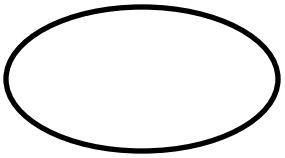
Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams

Attributes

Each entity type has a set of attributes associated w

An attribute is a property or characteristic of an enti
of interest to organization.



Attribute

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

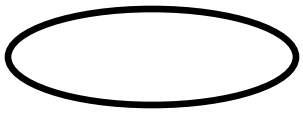
Entity-Relationship Diagrams

A candidate key is an attribute or combination of a
uniquely identifies each instance of an entity type.

Student_ID ⟶ Candidate Key

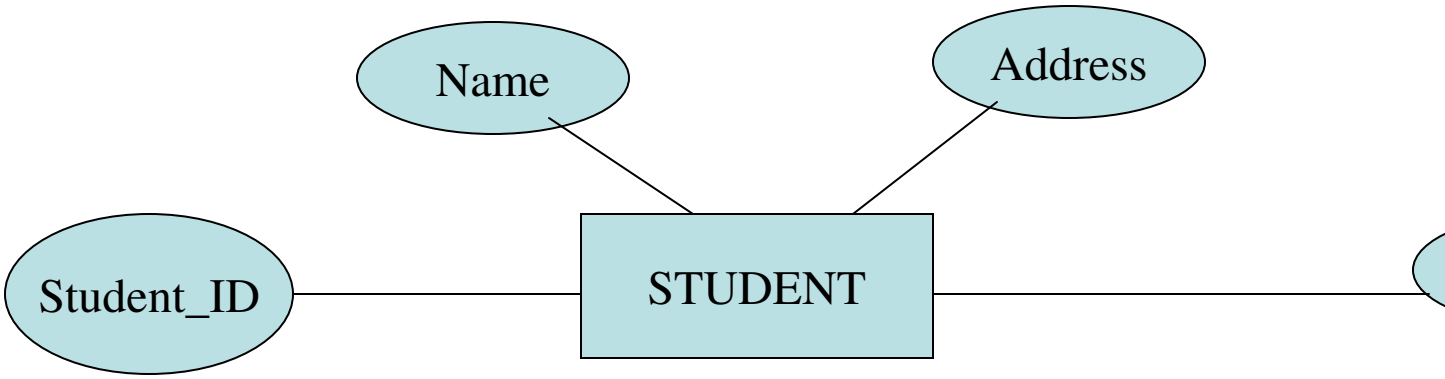
If there are more candidate keys, one of the key ma
as the Identifier.
It is used as unique characteristic for an entity type.

Identifier

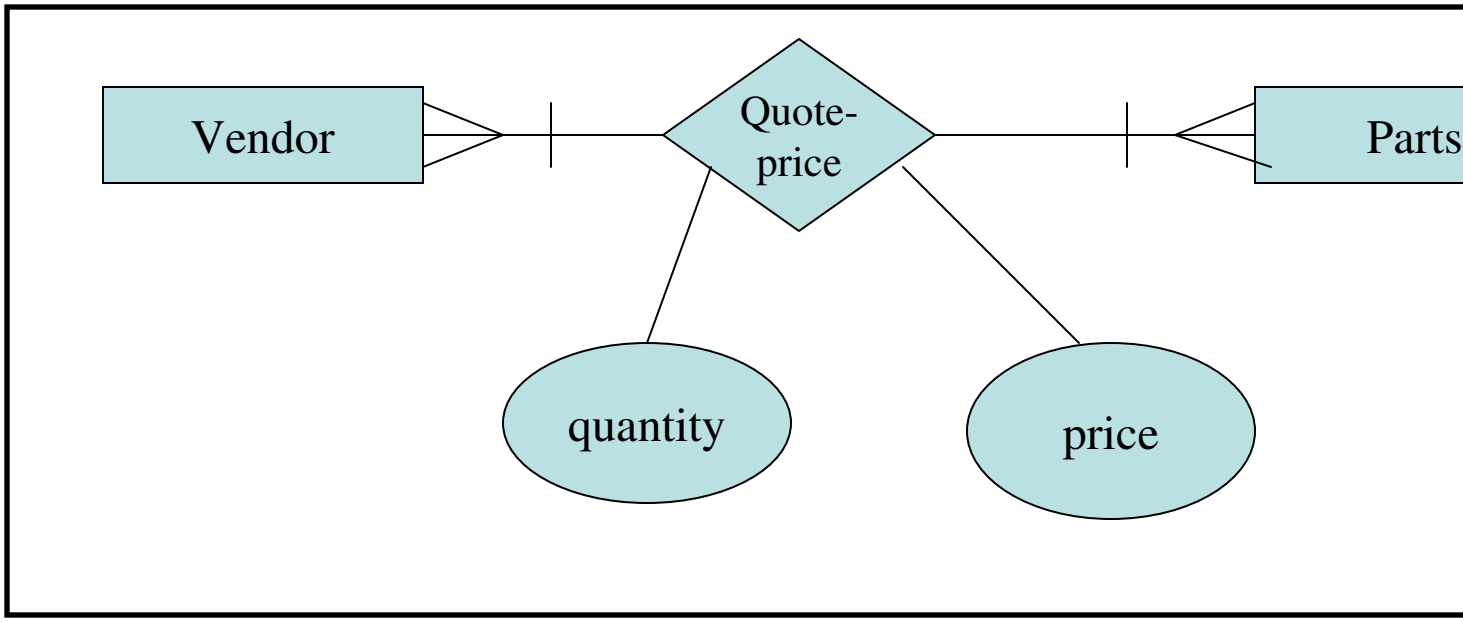


Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Entity-Relationship Diagrams



Vendors quote prices for several parts along with quantity of pa
Draw an E-R diagram.



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Approaches to problem analysis

1. List all inputs, outputs and functions.
2. List all functions and then list all inputs and ou
associated with each function.

Structured requirements definition (SRD)

Step1

Define a user level DFD. Record the inputs and
for each individual in a DFD.

Step2

Define a combined user level DFD.

Step3

Define application level DFD.

Step4

Define application level functions.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

This is the way of representing requirements in a consistent format

SRS serves many purpose depending upon who i it.

-- written by customer
-- written by developer

Serves as contract between customer & develope

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Nature of SRS

Basic Issues

- Functionality
- External Interfaces
- Performance
- Attributes
- Design constraints imposed on an Impleme

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

SRS Should

-- Correctly define all requirements

-- not describe any design details

-- not impose any additional constraints

Characteristics of a good SRS

An SRS Should be

✓ Correct

✓ Unambiguous

✓ Complete

✓ Consistent

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

- ✓ Ranked for important and/or stability
- ✓ Verifiable
- ✓ Modifiable
- ✓ Traceable

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

Correct

An SRS is correct if and only if every req
stated therein is one that the software shall meet.

Unambiguous

An SRS is unambiguous if and only
requirement stated therein has only one interpretatio

Complete

An SRS is complete if and only if, it incl
following elements

(i) All significant requirements, whether relate
functionality, performance, design co
attributes or external interfaces.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

(ii) Responses to both valid & invalid inputs.

(iii) Full Label and references to all figures, tables a
in the SRS and definition of all terms and units of m

Consistent

An SRS is consistent if and only if, no subset
individual requirements described in it conflict.

Ranked for importance and/or Stability

If an identifier is attached to every requireme
indicate either the importance or stability of that par
requirement.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

Verifiable

An SRS is verifiable, if and only if, every requi
stated therein is verifiable.

Modifiable

An SRS is modifiable, if and only if, its structur
are such that any changes to the requirements can b
easily, completely, and consistently while retaining st
style.

Traceable

An SRS is traceable, if the origin of each of the
requirements is clear and if it facilitates the referencir
requirement in future development or enhancement
documentation.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

Organization of the SRS

IEEE has published guidelines and standards to
SRS.
First two sections are same. The specific tailorin
section-3.

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Definition, Acronyms and abbre
    - 1.4 References
    - 1.5 Overview

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

2. The Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communication Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8 Site Adaptation Requirements

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Documentation

2.2 Product Functions
2.3 User Characteristics
2.4 Constraints
2.5 Assumptions for dependencies
2.6 Apportioning of requirements

3. Specific Requirements
   3.1 External Interfaces
   3.2 Functions
   3.3 Performance requirements
   3.4 Logical database requirements
   3.5 Design Constraints
   3.6 Software System attributes
   3.7 Organization of specific requirements
   3.8 Additional Comments.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

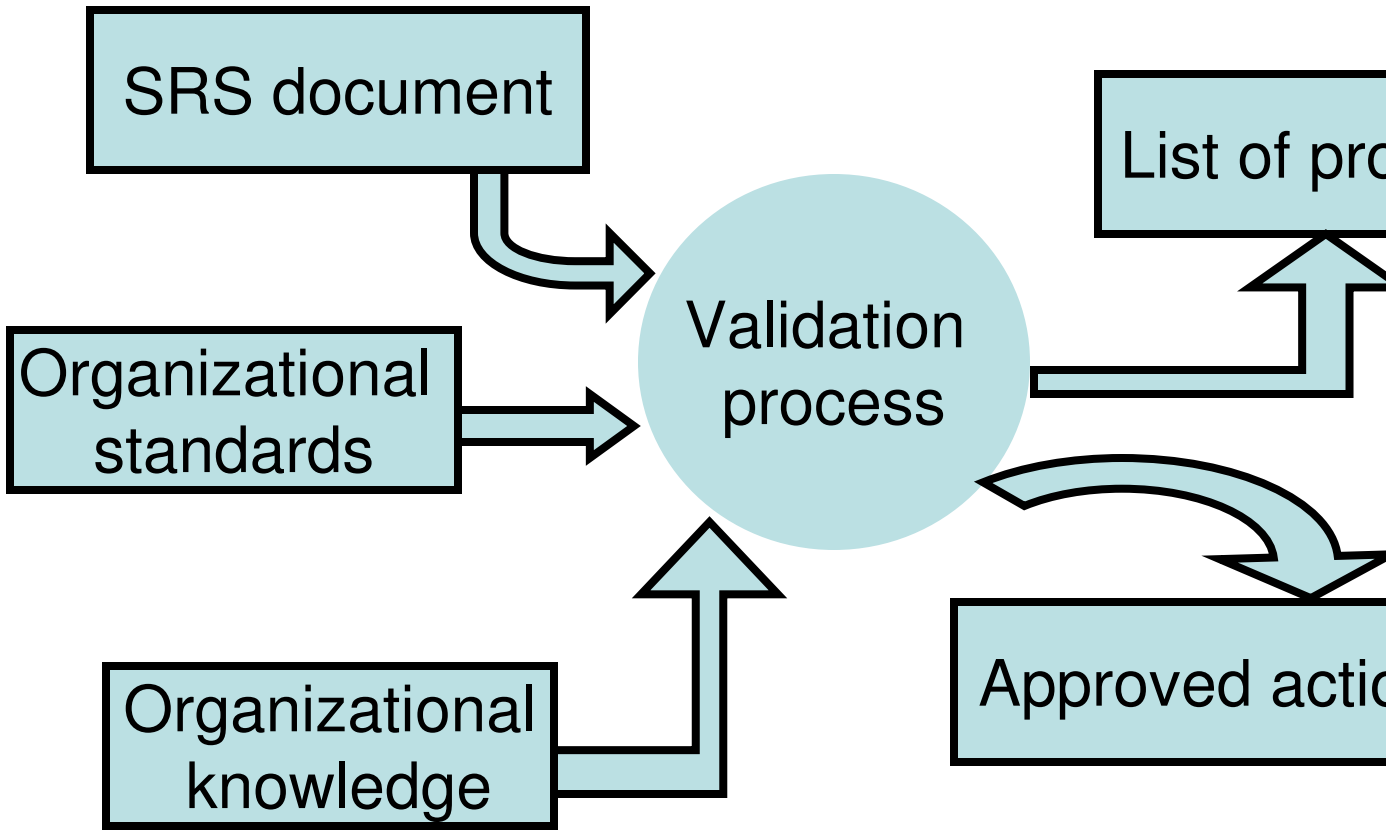
Requirements Validation

Check the document for:

- ✓ Completeness & consistency
- ✓ Conformance to standards
- ✓ Requirements conflicts
- ✓ Technical errors
- ✓ Ambiguous requirements

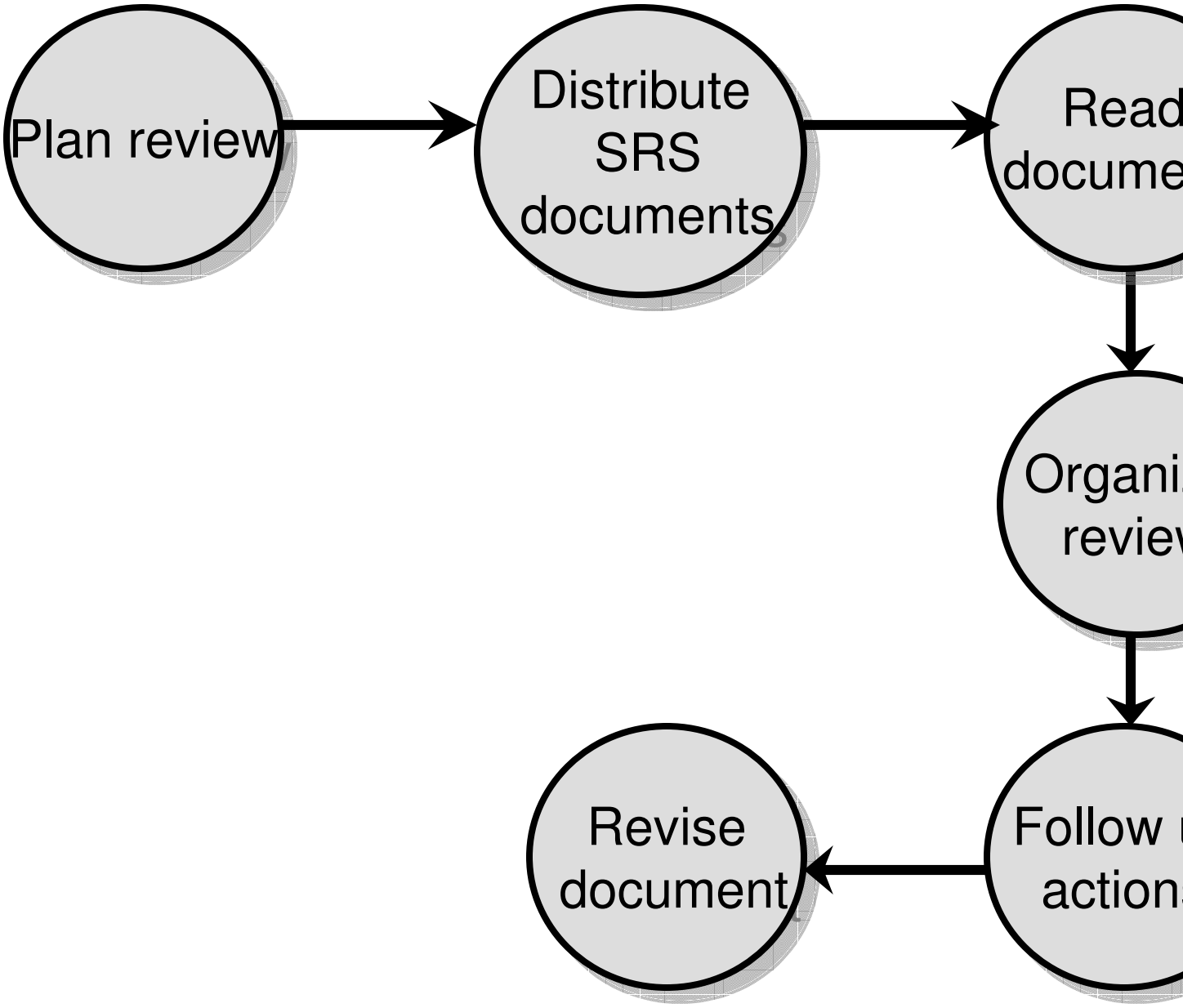
Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Validation



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Review Process



Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Validation

Problem actions

- Requirements clarification
- Missing information
  - find this information from stakeholders
- Requirements conflicts
  - Stakeholders must negotiate to resolve this conflict
- Unrealistic requirements
  - Stakeholders must be consulted
- Security issues
  - Review the system in accordance to security standards

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Review Checklists

- ✓ Redundancy
- ✓ Completeness
- ✓ Ambiguity
- ✓ Consistency
- ✓ Organization
- ✓ Conformance
- ✓ Traceability

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Prototyping

Validation prototype should be reasonably com
efficient & should be used as the required system

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Management

- Process of understanding and controlling cha system requirements.

ENDURING & VOLATILE REQUIREMENTS

o Enduring requirements: They are core requiren are related to main activity of the organization.

  Example: issue/return of a book, cataloging etc.

o Volatile requirements: likely to change during s development lifer cycle or after delivery of the pro

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Management Planning

- Very critical.
- Important for the success of any project.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Requirements Change Management

- Allocating adequate resources
- Analysis of requirements
- Documenting requirements
- Requirements traceability
- Establishing team communication
- Establishment of baseline

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Download from

Q:\IRM\PRIVATE\INITIATIATI\QA\QAPLAN\SRSPLAN.DOC

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

Note. Choose the most appropriate answer of the following questions.

3.1 Which one is not a step of requirement engine

(a) Requirements elicitation
(b) Requirements analysis
(c) Requirements design
(d) Requirements documentation

3.2 Requirements elicitation means

(a) Gathering of requirements
(b) Capturing of requirements
(c) Understanding of requirements
(d) All of the above

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.3 SRS stands for

(a) Software requirements specification
(b) System requirements specification
(c) Systematic requirements specificatior
(d) None of the above

3.4 SRS document is for

(a) "What" of a system?
(b) How to design the system?
(c) Costing and scheduling of a system
(d) System's requirement.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.5 Requirements review process is carried out to

(a) Spend time in requirements gathering
(b) Improve the quality of SRS
(c) Document the requirements
(d) None of the above

3.6 Which one of the statements is not correct dur requirements engineering?

(a) Requirements are difficult to uncover
(b) Requirements are subject to change
(c) Requirements should be consistent
(d) Requirements are always precisely known

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.7 Which one is not a type of requirements?

(a) Known requirements
(b) Unknown requirements
(c) Undreamt requirements
(d) Complex requirements

3.8 Which one is not a requirements elicitation technique?

(a) Interviews
(b) The use case approach
(c) FAST
(d) Data flow diagram.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.9 FAST stands for

(a) Functional Application Specification Technique
(b) Fast Application Specification Technique
(c) Facilitated Application Specification Technique
(d) None of the above

3.10 QFD in requirement engineering stands for

(a) Quality function design
(b) Quality factor design
(c) Quality function development
(d) Quality function deployment

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.11 Which is not a type of requirements under quality function deployment

(a) Normal requirements
(b) Abnormal requirements
(c) Expected requirements
(d) Exciting requirements

3.12 Use case approach was developed by

(a) I. Jacobson and others
(b) J.D. Musa and others
(c) B. Littlewood
(d) None of the above

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.13 Context diagram explains

(a) The overview of the system
(b) The internal view of the system
(c) The entities of the system
(d) None of the above

3.14 DFD stands for

(a) Data Flow design
(b) Descriptive functional design
(c) Data flow diagram
(d) None of the above

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.15 Level-O DFD is similar to

(a) Use case diagram
(b) Context diagram
(c) System diagram
(d) None of the above

3.16 ERD stands for

(a) Entity relationship diagram
(b) Exit related diagram
(c) Entity relationship design
(d) Exit related design

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.17 Which is not a characteristic of a good SRS?

(a) Correct
(b) Complete
(c) Consistent
(d) Brief

3.18 Outcome of requirements specification phase

(a) Design Document
(b) Software requirements specification
(c) Test Document
(d) None of the above

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.19 The basic concepts of ER model are:

(a) Entity and relationship
(b) Relationships and keys
(c) Entity, effects and relationship
(d) Entity, relationship and attribute

3.20 The DFD depicts

(a) Flow of data
(b) Flow of control
(c) Both (a) and (b)
(d) None of the above

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.21 Product features are related to:

(a) Functional requirements
(b) Non functional requirements
(c) Interface requirement
(d) None of the above

3.22 Which one is a quality attribute?

(a) Reliability
(b) Availability
(c) Security
(d) All of the above

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

3.23 IEEE standard for SRS is:

(a) IEEE Standard 837-1998
(b) IEEE Standard 830-1998
(c) IEEE Standard 832-1998
(d) IEEE Standard 839-1998

3.24 Which one is not a functional requirement

(a) Efficiency
(b) Reliability
(c) Product features
(d) Stability

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

3.23 APIs stand for:

(a) Application performance interfaces
(b) Application programming interfaces
(c) Application programming integration
(d) Application performance integration

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Exercises

3.1 Discuss the significance and use of requirement engineer the problems in the formulation of requirements?

3.2 Requirements analysis is unquestionably the most c intensive step in the software engineering process. W communication path frequently break down ?

3.3 What are crucial process steps of requirement engineering ' the help of a diagram.

3.4 Discuss the present state of practices in requirement engine few steps to improve the present state of practice.

3.5 Explain the importance of requirements. How ma requirements are possible and why ?

3.6 Describe the various steps of requirements engineering. Is follow these steps ?

3.7 What do you understand with the term "requirements Discuss any two techniques in detail.

3.8 List out requirements elicitation techniques. Which one is and why ?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Exercises

3.9 Describe facilitated application specification technique compare this with brainstorming sessions.

3.10 Discuss quality function deployment technique of elicitation. Why an importance or value factor is associate requirement ?

3.11. Explain the use case approach of requirements elicitat use-case guidelines ?

3.12. What are components of a use case diagram. Explain thei the help of an example.

3.13. Consider the problem of library management system and following:

(*i*) Problem statement

(*ii*) Use case diagram

(*iii*) Use cases.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Exercises

3.14. Consider the problem of railway reservation system and d
following:

(*i*) Problem statement

(*ii*) Use case diagram

(*iii*) Use cases.

3.15. Explain why a many to many relationship is to be modele
associative entity ?

3.16. What are the linkages between data flow and E–R diagran

3.17. What is the degree of a relationship ? Give an example of
relationship degree.

3.18. Explain the relationship between minimum cardinality an
mandatory participation.

3.19. An airline reservation is an association between a passeng
and a seat. Select a few

pertinent attributes for each of these entity types and represent a
in an E–R diagram.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Exercises

3.20. A department of computer science has usual resources a
for these resources. A software is to be developed so that
assigned without conflict. Draw a DFD specifying the above sy

3.21. Draw a DFD for result preparation automation system
courses (or MCA program) of any university. Clearly describ
of the system. Also mention all assumptions made by you.

3.22. Write short notes on

(*i*) Data flow diagram

(*ii*) Data dictionary.

3.23. Draw a DFD for borrowing a book in a library which
below: "A borrower can borrow a book if it is available el
reserve for the book if he/she so wishes. He/she can borrow a
three books".

3.24. Draw the E–R diagram for a hotel reception desk manage

Explain why, for large software systems development, is it
that prototypes should be "throw-away" prototype ?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Exercises

3.26. Discuss the significance of using prototyping for reusabl and explain the problems,which may arise in this situation.

3.27. Suppose a user is satisfied with the performance of a he/she is interested to buy this

for actual work, what should be the response of a developer ?

3.28. Comment on the statement: "The term throw-away inappropriate in that these prototypes expand and enhance t base that is retained and incorporated in the final prototype; are not disposed of or thrown away at all."

3.29. Which of the following statements are ambiguous ? Expla

(*a*) The system shall exhibit good response time.

(*b*) The system shall be menu driven.

(*c*) There shall exist twenty-five buttons on the control panel, r to PF25.

(*d*) The software size shall not exceed 128K of RAM.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Exercises

3.30. Are there other characteristics of an SRS (besides listed in
3.4.2) that are desirable ? List a few and describe why ?

3.31. What is software requirements specification (SRS) ? List
advantages of SRS standards.

Why is SRS known as the black box specification of a system ?

3.32. State the model of a data dictionary and its contents. Wha
advantages ?

3.33. List five desirable characteristics of a good SRS documen
relative advantages of formal requirement specifications. List th
issues, which an SRS must address.

3.34. Construct an example of an inconsistent (incomplete) SRS

3.35. Discuss the organization of a SRS. List out some importan
this organization.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Exercises

3.36. Discuss the difference between the following:

(a) Functional & nonfunctional requirements

(b) User & system requirements

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007