# Software Requirement Concepts & Process

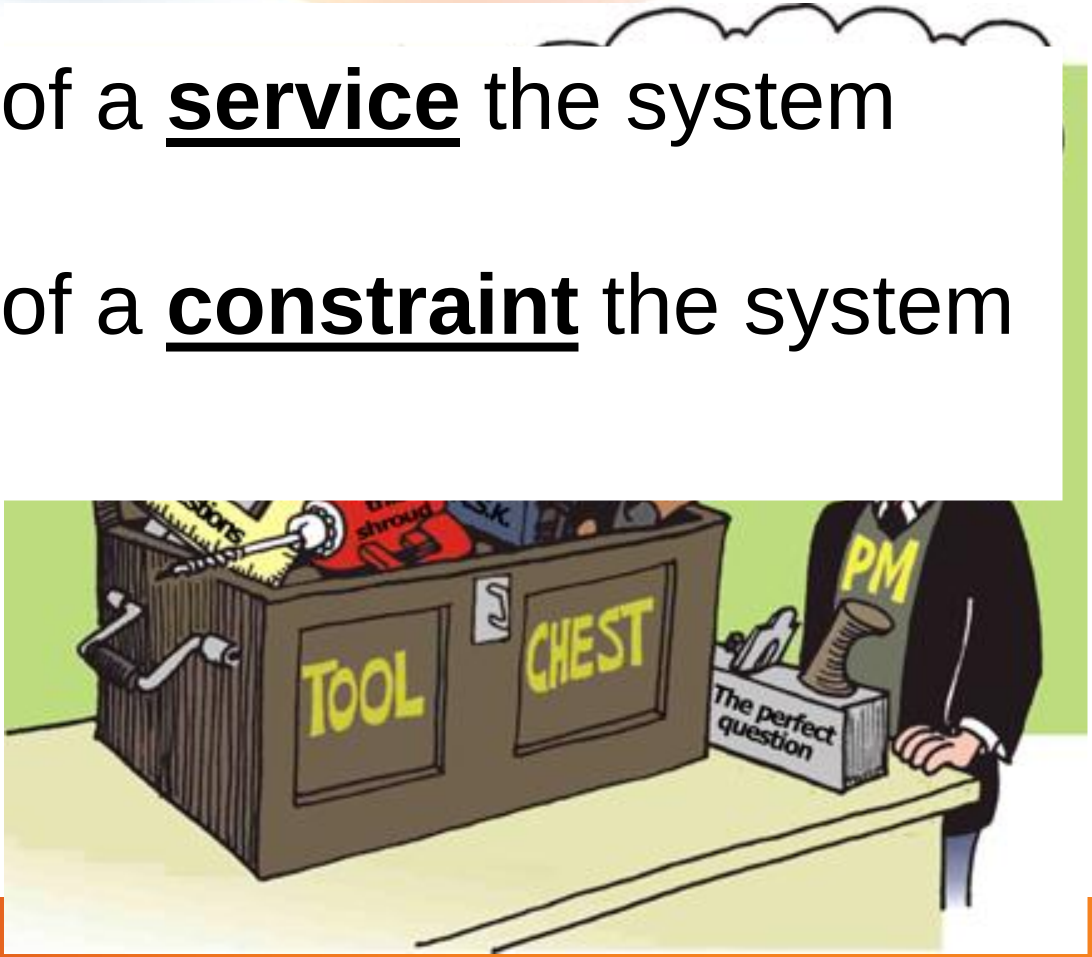
# Agenda

1. Requirement Concepts
2. Fsoft Requirement Process
3. Requirement Clarifying
4. Requirement Modeling
5. Modeling Tools
6. Common practices, problems

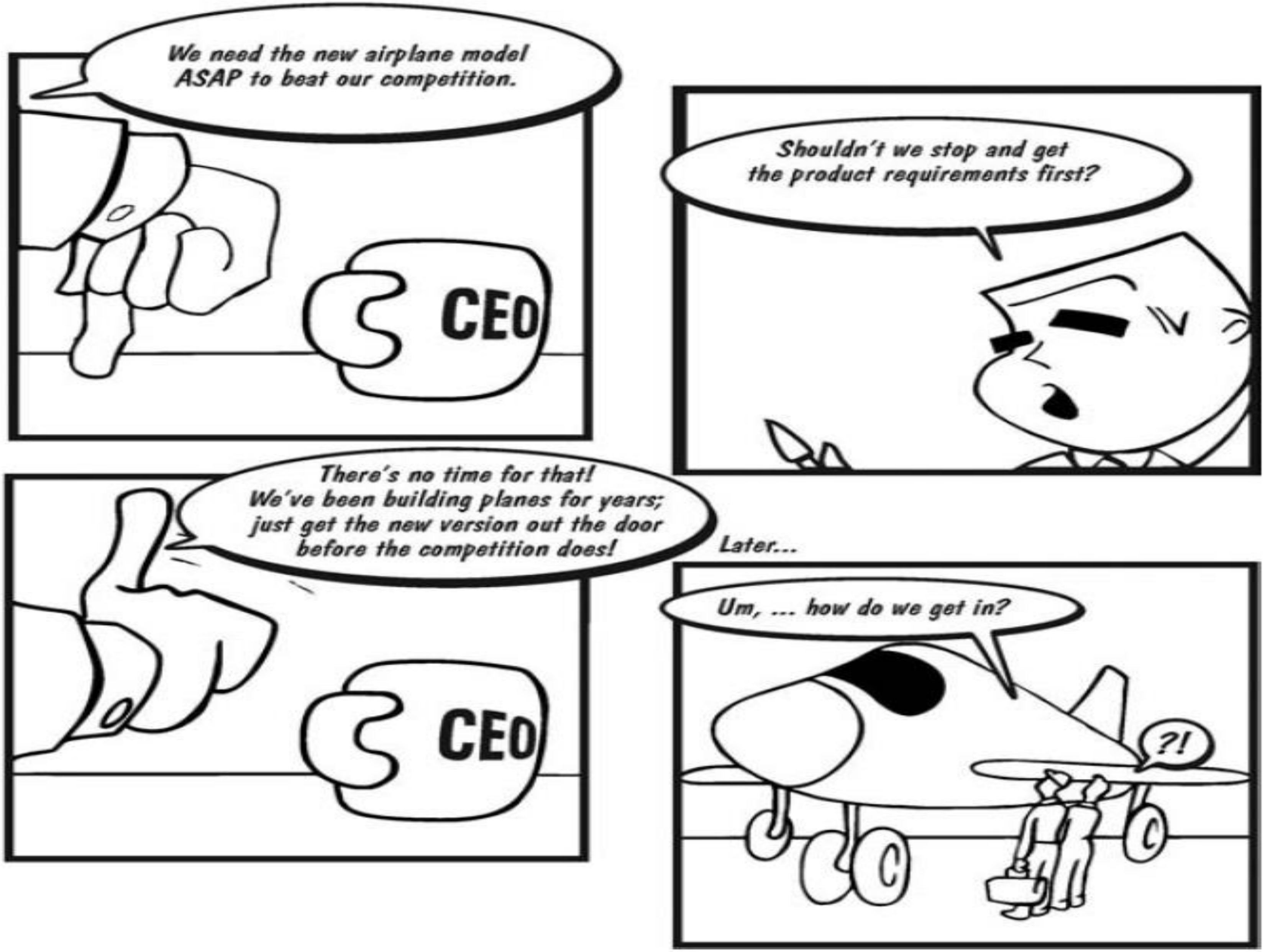
# Requirement Concepts
# Requirement Definition

- What is requirement?

- A statement of a **service** the system must do OR
- A statement of a **constraint** the system must satisfy



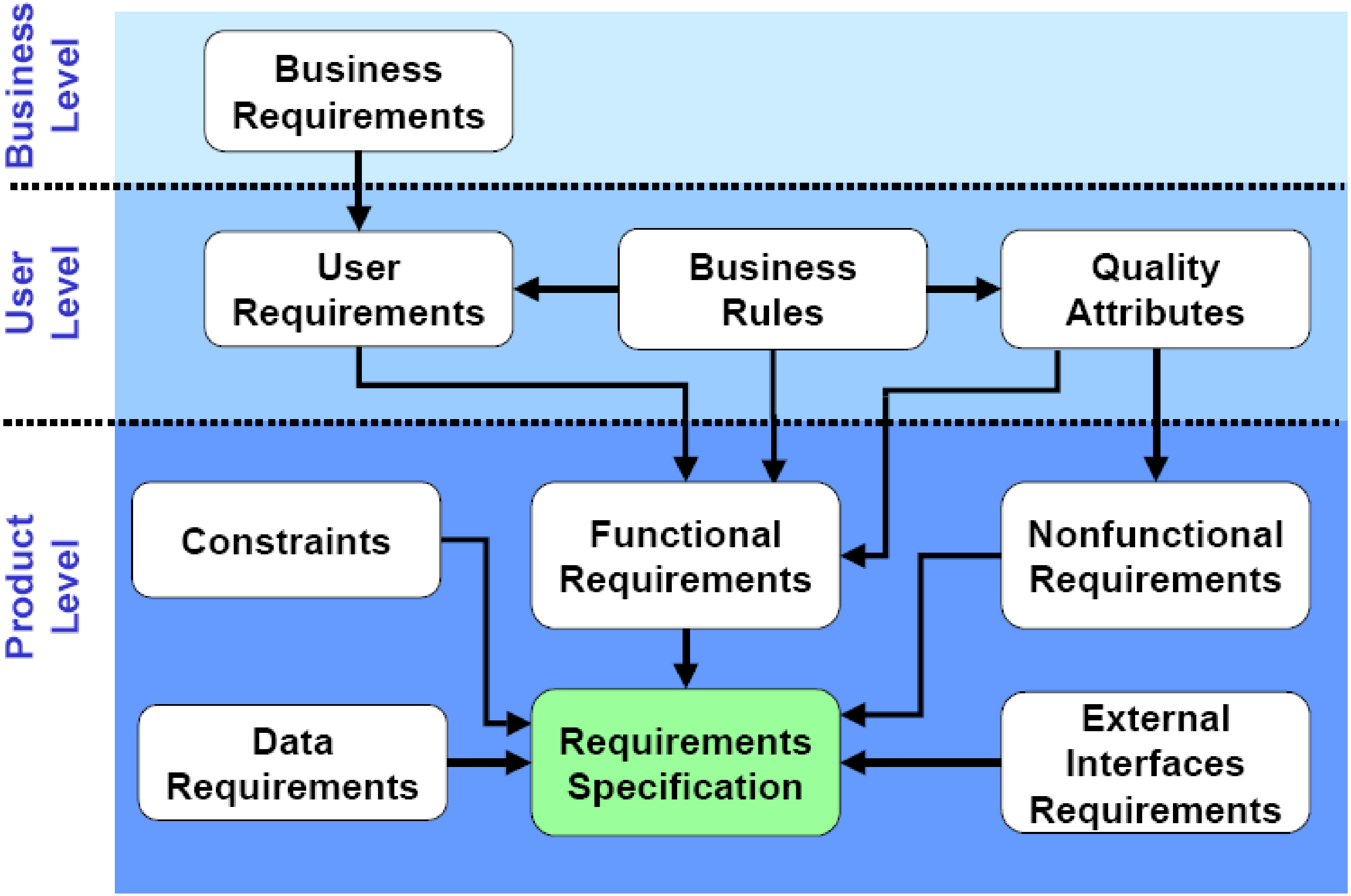
# Requirement Concepts
# Requirement Definition


We need the new airplane model ASAP to beat our competition.
CEO
Shouldn't we stop and get the product requirements first?
There's no time for that! We've been building planes for years; just get the new version out the door before the competition does!
CEO
Later...
Um, ... how do we get in?
?!

# Requirement Concepts
# Requirement Definition

- Purpose of requirement:
  - Requirements often serve as:
    - The basis for a bid for a contract - therefore must be **high-level** to open for interpretation
    - The basis for the contract itself - therefore must be **detailed**
  - Thus, requirements can be **high-level** or **detailed**
- What are not Requirements
  - Design or implementation details (other than known constraints)
  - Project planning information
  - Testing information

# Requirement Concepts
# Requirements Classification 1/4



# Requirement Concepts
# Requirements Classification 2/4

- Requirement may be classified as
  - Functional
    - A **service** the system has to perform
    - May include information the system must contain
  - Non-functional
    - A **constraints** the system must satisfy

# Requirement Concepts
# Requirements Classification 3/4

- Sample of functional requirement

  The “Data Entry Module” should provide the following functionality:

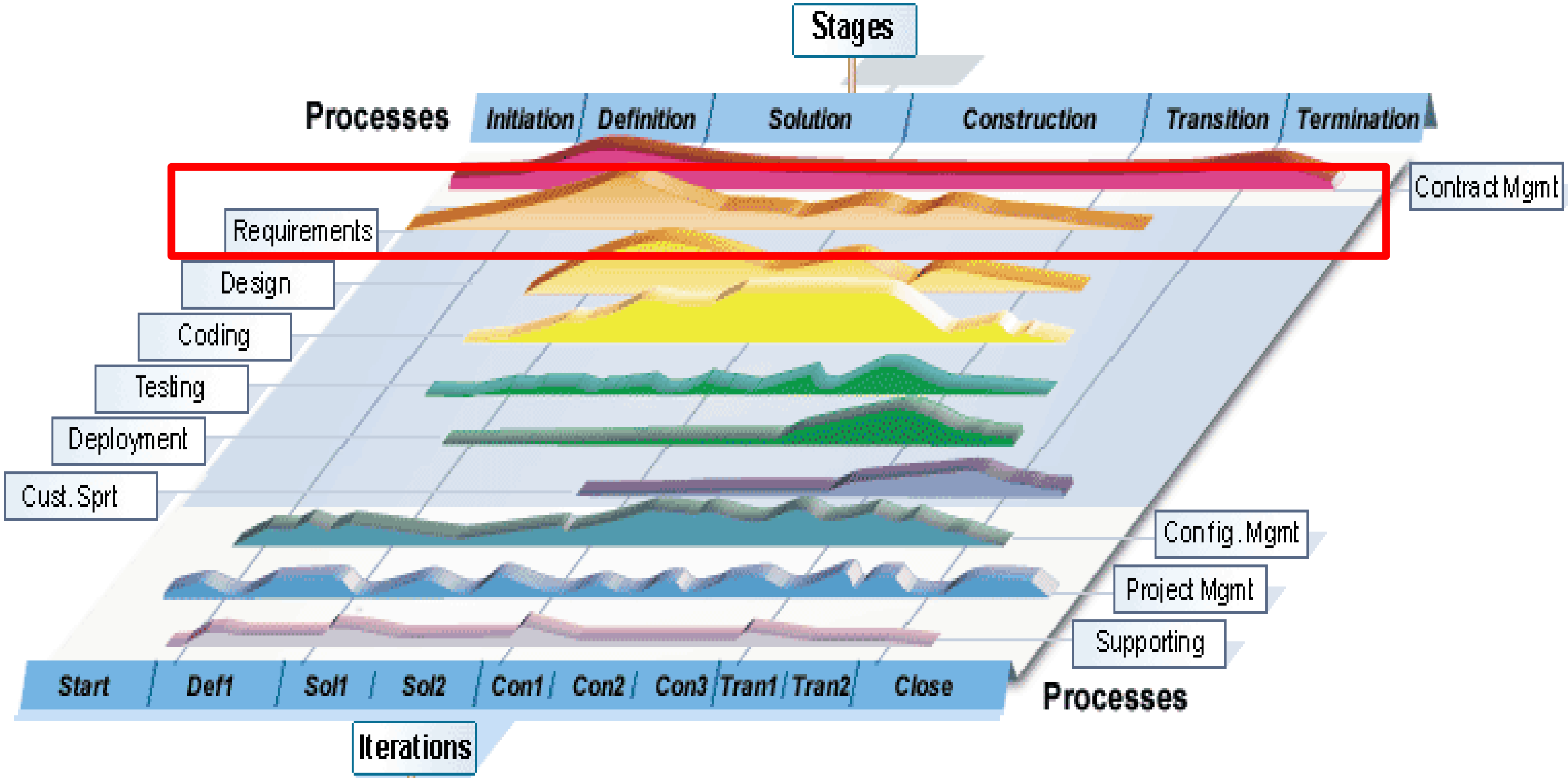
  - **Data Entry for HR**: allows HR staff to enter payroll data, either via web-based forms or by importing data from Excel files
  - **Data Entry for Regional offices**:  allows the PGB’s regional offices to enter billing data, either via web-based forms or by importing data from Excel files

# Requirement Concepts
# Requirements Classification 4/4

- Sample of non-functional requirement
  - Product requirements
    - – Requirements which specify that the delivered product must behave in a particular way
    - – Categories: performance, reliability, usability, security, cultural, etc.
  - Organisational requirements
    - – Requirements which are a consequence of organisational policies and procedures
    - – Categories : technology, process, operation, time, budget, etc.
  - External requirements
    - – Requirements which arise from factors which are external to the system and its development process
    - – Categories : interoperability requirements, legislative requirements, etc.

# Fsoft Requirement Process

- First phase of Software engineering



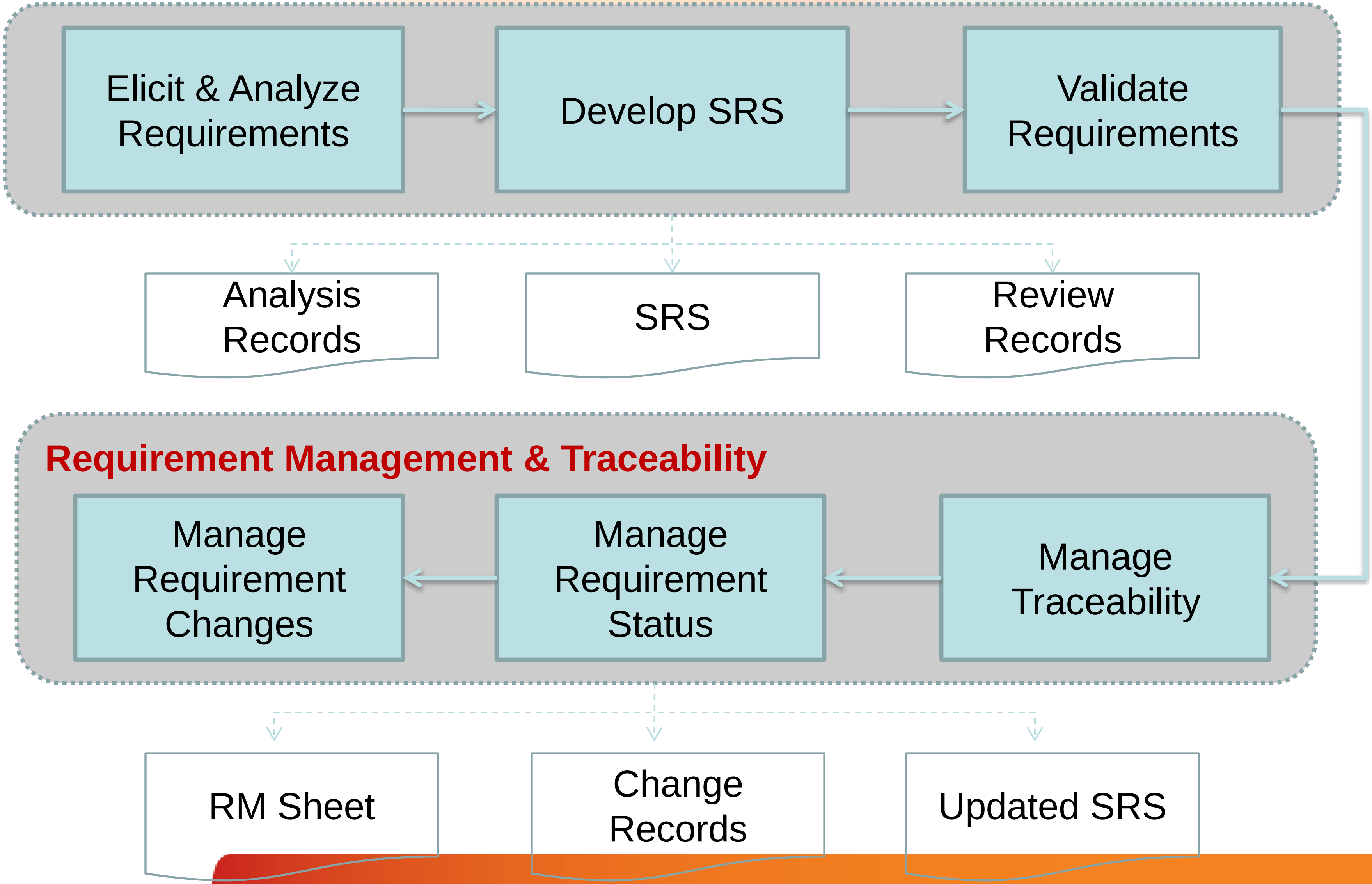
# Fsoft Requirement Process

## Objectives

- To ensure that requirements for the software product are defined and understood.
  - Get to know what customer's requirement is
  - Understand the customers' needs & expectation
- To create SRS - Establish and maintain requirements agreement with the requestor and affected groups
- To ensure that the requirements are met.
- Requirements are documented and controlled to establish a basis for software development and project management use.

# Fsoft Requirement Process

## Workflow



Elicit & Analyze Requirements
Develop SRS
Validate Requirements
Analysis Records
SRS
Review Records
Requirement Management & Traceability
Manage Requirement Changes
Manage Requirement Status
Manage Traceability
RM Sheet
Change Records
Updated SRS

# Fsoft Requirement Process

## Elicit & Analyze Requirements

- Sometimes called **requirements discovery**
- Requirements are often not given to you, you have to **elicit** them; Must work with **customers** and relevant **stakeholders** to elicit:
  - the **services** that the system should provide
  - the **constraints** that the system should satisfy
- Requirement analysis is done to:
  - Detect and resolve conflicts between requirements
  - Discover the bounds of the software and how it must interact with its environment
  - Elaborate system requirements to derive software requirements

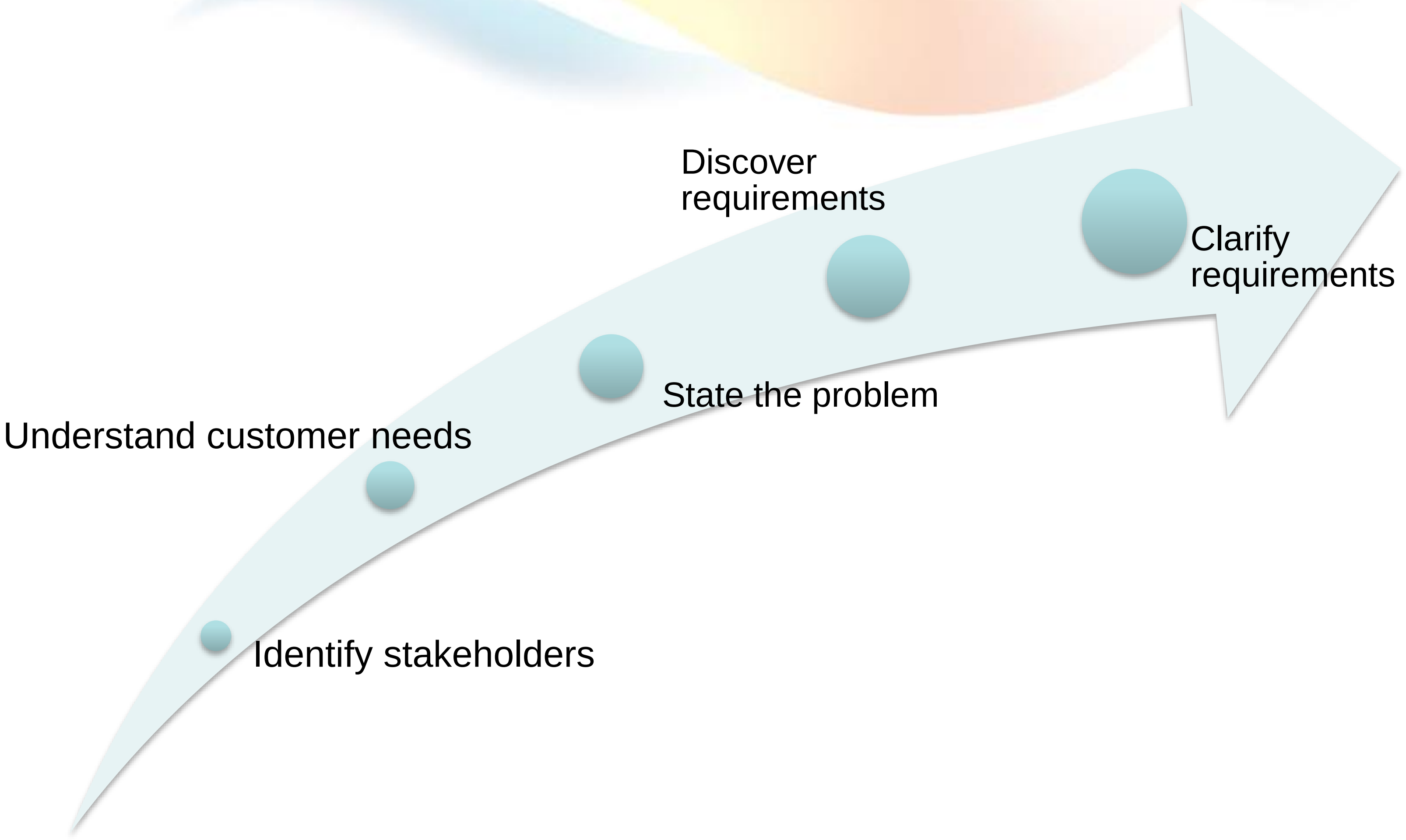
# Fsoft Requirement Process

## Elicit & Analyze Requirements - Resource

- Potential **stakeholders**
  - End-users
  - Managers
  - Owners
  - Customers of your customers
  - Operation engineers
  - Domain experts
  - Trade unions
  - Etc.

# Fsoft Requirement Process

## Elicit & Analyze Requirements - Process



# Fsoft Requirement Process

## Elicit & Analyze Requirements - Issues

- Issues of scope
  - – The boundary of the system is ill-defined
  - – The customers/users specify unnecessary technical detail that may confuse overall system objectives
- Issues of understanding
  - – The customers/users are not completely sure of what is needed
  - – Have a poor understanding of the capabilities and limitations of their computing environment
  - – Don’t have a full understanding of the problem domain, have trouble communicating needs to the system engineer

# Fsoft Requirement Process

## Elicit & Analyze Requirements - Issues

- Issues of understanding (cont’d)
  - Omit information that is believed to be “obvious”
  - Specify requirements that conflict with the needs of other customers/users
  - Specify requirements that are ambiguous or un-testable.
- Issues of volatility
  - The requirements change over time

# Fsoft Requirement Process

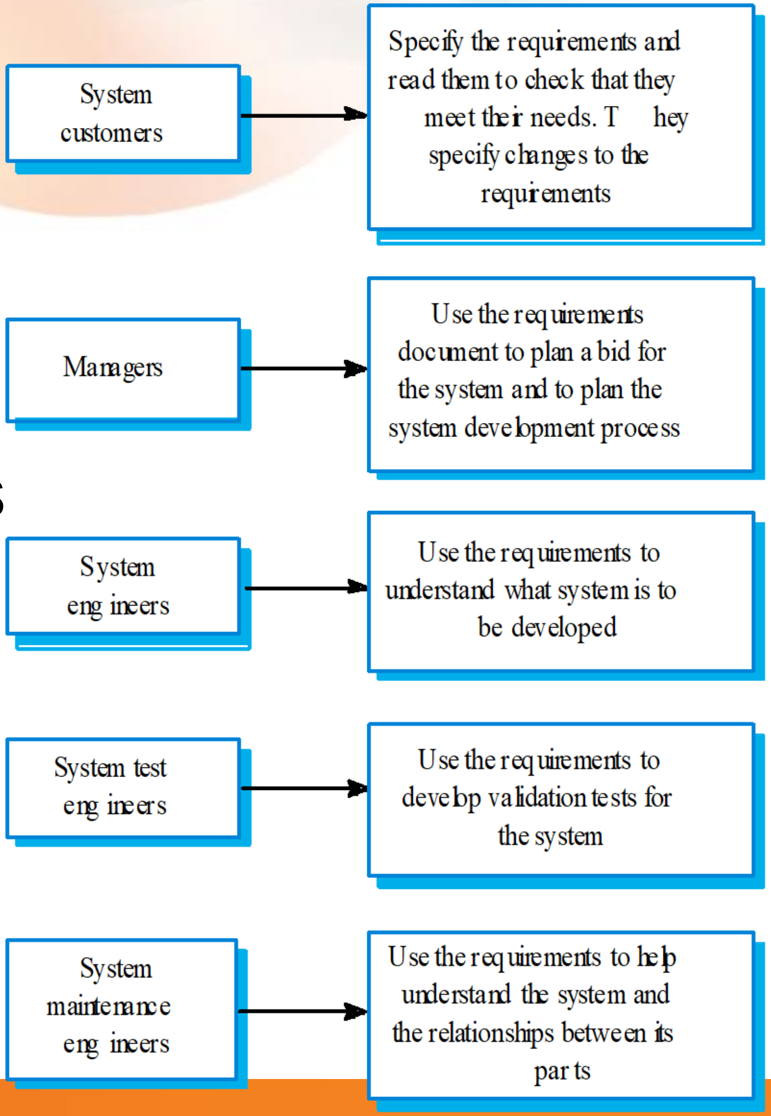
## Elicit & Analyze Requirements - Techniques

- Elicitation Techniques
  - Researching application domain
  - Interviewing and questionnaires
  - Workshop and brainstorming
  - Storyboarding and role playing
  - Observation
  - Use cases
- Analyzing Techniques
  - System modeling
  - Rapid Prototyping

# Fsoft Requirement Process

## Develop SRS - Requirement documents 1/3

- Requirements document is the **official document** of what is required for the system
- Often include only **system requirements** but sometimes may also include **user requirements**
- It is **NOT** a design document. Describe **WHAT** the system should do rather than **HOW** it should do



System customers → Specify the requirements and read them to check that they meet their needs. They specify changes to the requirements

Managers → Use the requirements document to plan a bid for the system and to plan the system development process

System engineers → Use the requirements to understand what system is to be developed

System test engineers → Use the requirements to develop validation tests for the system

System maintenance engineers → Use the requirements to help understand the system and the relationships between its parts

# Fsoft Requirement Process

## Develop SRS - Requirement documents 2/3

- URD – User requirement definition
  - Address what users need to do their jobs
  - Composed all business requirements formulated by customer, business rules and other constrains
- SRS – Software requirement specification
  - A set of software requirements as complete, consistent, and correct as possible, from the developer's point of view
  - Document which after baselining, common reference point of the software requirements for customer, developer, tester and PM .

# Fsoft Requirement Process

## Develop SRS - Requirement documents 3/3

- Benefit of good document
  - Basis for agreement between the customers and the team on what the software product is to do.
  - Reduce the development effort.
  - Provide a basis for estimating costs, schedules.
  - Provide a baseline for validation and verification.
  - Facilitate transfer.
  - Serve as a basis for enhancement

# Fsoft Requirement Process

## Develop SRS – Steps & Activities

- Study URD:
- Analyze user requirement
- Prepare Q&A list to clarify unclear items with customers
- Call/interview customers if needed
- SRS:
- Develop use cases, system requirement
- Develop functional specification
- Review and approve SRS:
- Call up meeting for review
- Keep meeting minutes records

# Fsoft Requirement Process

## Develop SRS – Techniques

- Specify requirements using **structured natural language** (forms, tables, etc.)
- **Functional requirements** can be specified using modeling - a combination of graphical notations and structured natural language
  - Use cases
  - Use case diagrams
  - Use case specifications
- **Non-functional requirements** can't be modeled => specified using structured natural language only

# Fsoft Requirement Process

## Develop SRS - Characteristics of good SRS

- Correct: requirement ~ what the software shall meet.
- Unambiguous:
  - Has only one interpretation (to both creator & user)
  - Use natural language & avoid the words like: maybe, generally, etc.
- Complete
  - Include all significant requirements.
  - Define all the software responses & include all the refs/labels.
  - Use of TBD: should avoid OR mention why, what to do, who, when.
- Consistent: no conflict between individual requirements.
- Verifiable: reviewable & testable in finite cost-effective process.
- Traceable: clear origin & good reference for future dev/enhance documents.

# Fsoft Requirement Process

## Develop SRS - SRS Review Checklist

- SRS Review Checklist
  - To review the requirements by yourself
  - Make sure you understood completely the requirements:
    - Organization and Completeness: adequate, no missing, etc.
    - Correctness: no conflict, verifiable, in scope, message, etc.
    - Non-functional requirements, quality attributes, etc.
  - Template (See Checklist - Review Requirement.xlsx)

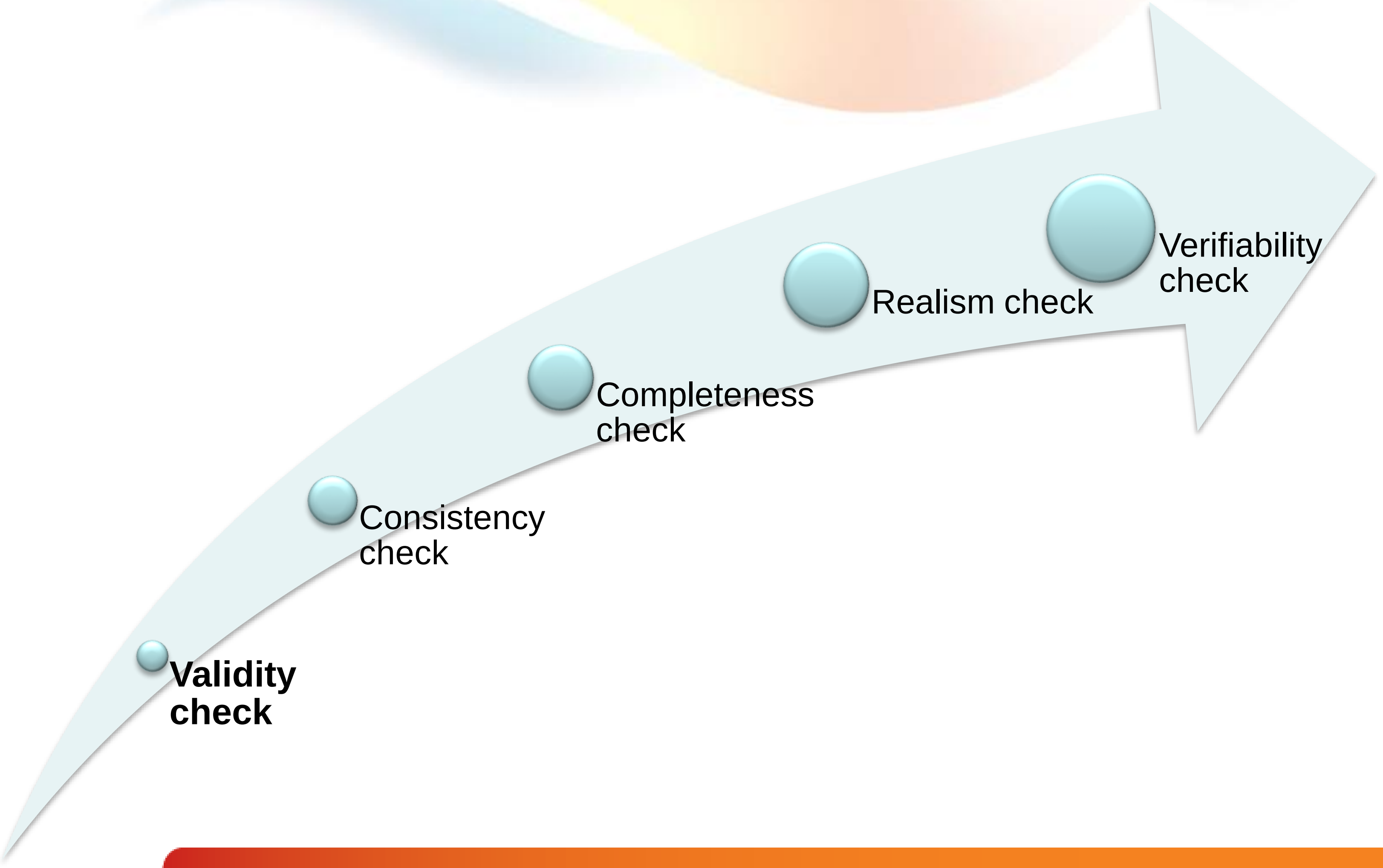
# Fsoft Requirement Process

## Validate Requirements – Purpose

- Make sure that the requirements define the system that the customer really wants
- Requirements error costs are high so validation is very important
  - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error

# Fsoft Requirement Process

## Validate Requirements – Process



**Validity check**

Consistency check

Completeness check

Realism check

Verifiability check

# Fsoft Requirement Process
## Validate Requirements – Techniques

- Requirements Review
  - Systematic manual analysis of the requirements
  - Involving development staff, customers and relevant stakeholders
- Prototyping
  - Using an executable model of the system to check requirements
- Model Validation
  - Validate the quality of the models developed during analysis
- Test-case generation
  - Developing tests for requirements to check testability

# Fsoft Requirement Process
## Requirements management

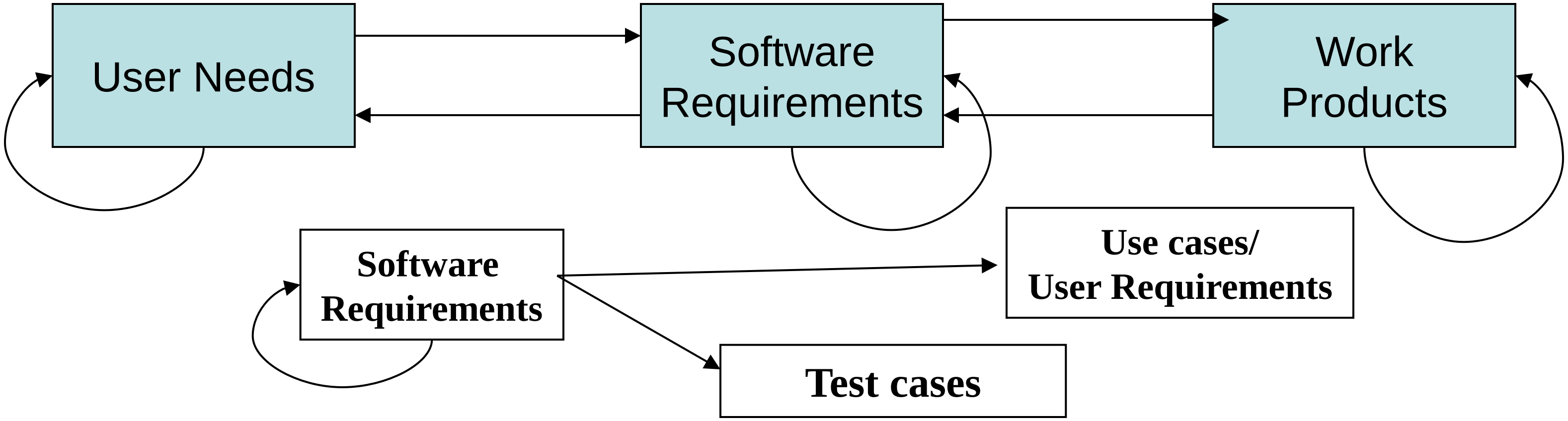
- Manage requirement in FSOFT
  - Requirement Management Sheet, Excel sheet, used to track the status, relationship and change of requirements during the whole project.
  - A mandatory document (dynamic version of SRS)
    - Classify requirement to functional/non-functional requirement
    - To maintain the common reference for all related parties (traceability of requirement and software product)
    - To track the project progress (status of requirement)
    - To track the change (including change request)
    - To collect requirement related metrics for reporting
  - The sheet is created the first time client requirement come

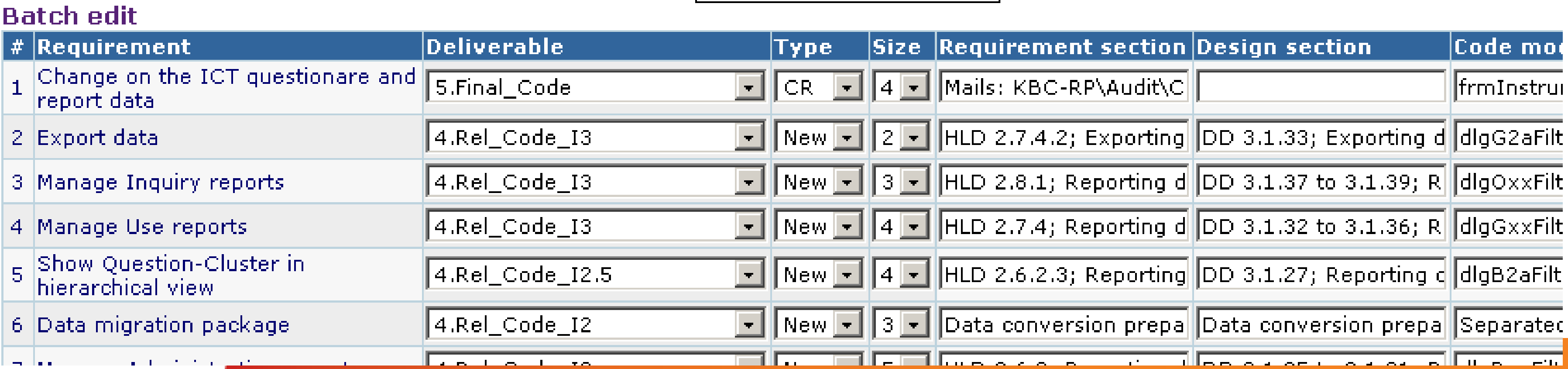
# Fsoft Requirement Process

## Requirement Traceability

Why is traceability necessary?

- The requirements can change at any stage during the product's life.
- If the requirements are traceable, then when changes happen, it is far easier to find the impacted parts of the product





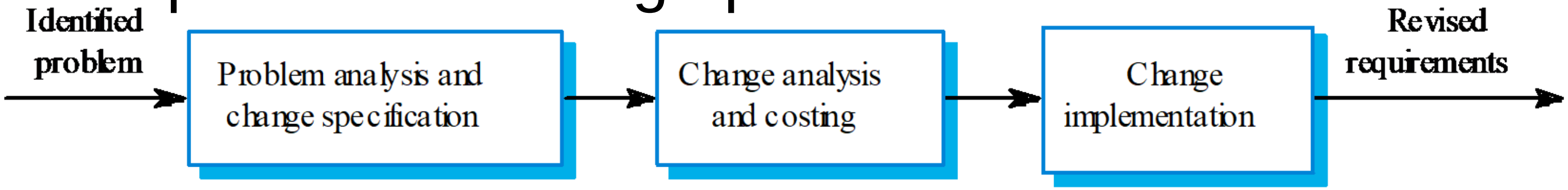
Batch edit

| # | Requirement | Deliverable | Type | Size | Requirement section | Design section | Code mo |
|---|---|---|---|---|---|---|---|
| 1 | Change on the ICT questionare and report data | 5.Final_Code | CR | 4 | Mails: KBC-RP\Audit\C |  | frmInstru |
| 2 | Export data | 4.Rel_Code_I3 | New | 2 | HLD 2.7.4.2; Exporting | DD 3.1.33; Exporting d | dlgG2aFilt |
| 3 | Manage Inquiry reports | 4.Rel_Code_I3 | New | 3 | HLD 2.8.1; Reporting d | DD 3.1.37 to 3.1.39; R | dlgOxxFilt |
| 4 | Manage Use reports | 4.Rel_Code_I3 | New | 4 | HLD 2.7.4; Reporting d | DD 3.1.32 to 3.1.36; R | dlgGxxFilt |
| 5 | Show Question-Cluster in hierarchical view | 4.Rel_Code_I2.5 | New | 4 | HLD 2.6.2.3; Reporting | DD 3.1.27; Reporting c | dlgB2aFilt |
| 6 | Data migration package | 4.Rel_Code_I2 | New | 3 | Data conversion prepa | Data conversion prepa | Separatec |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

# Fsoft Requirement Process

## Requirement Changes Management

- Requirements change (CR – Change request)
  - The priority of requirements from different viewpoints changes during the development process
  - Customers may specify requirements from a business perspective that conflict with end-user requirements
  - The business and technical environment of the system changes during its development
- Requirements change process


Identified problem
Problem analysis and change specification
Change analysis and costing
Change implementation
Revised requirements

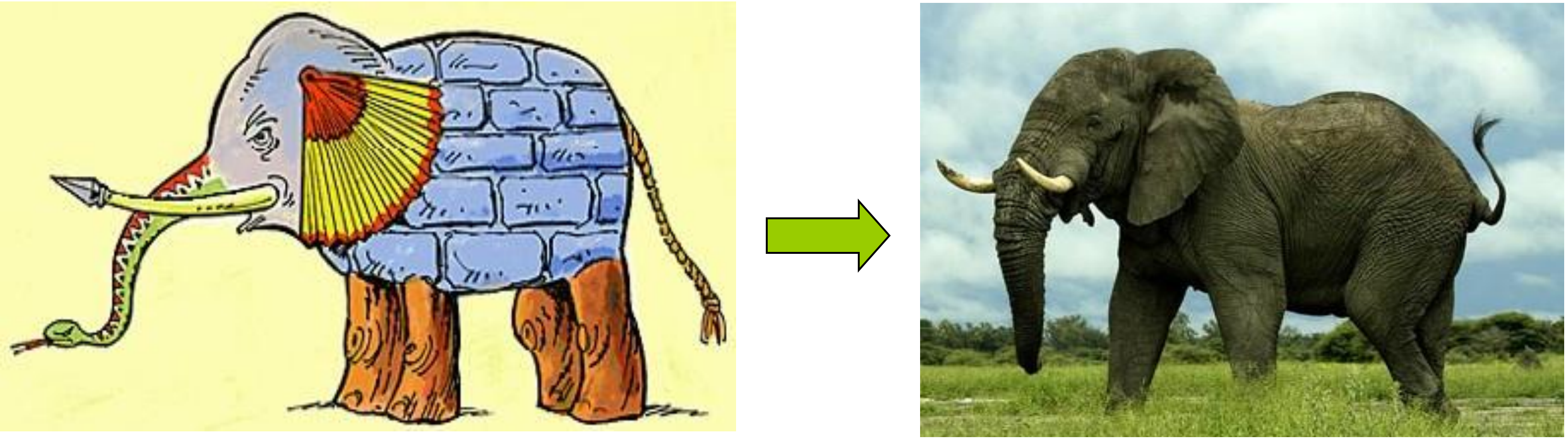
# Requirement Clarifying

- PM/TL/BA present the SRS to members in the team
- Self study related materials: top-down approach
- Using FSOFT’s SRS review checklist
- Clarify unclear item(s) using Q&A
- Discuss with other members
  - To clarify or confirm your understanding
  - Media: direct discussion or via team brainstorming
- Inform the PM/TL/BA about
  - Any requirement conflicts
  - Changes, comparing to the last version

# Requirement Clarifying

## Clarifying requirement via Q&A

- Why do we need Q&A?
  - Problems of understanding
  - Want for knowledge, must be ask



# Requirement Clarifying

## Clarifying requirement via Q&A (cont.)

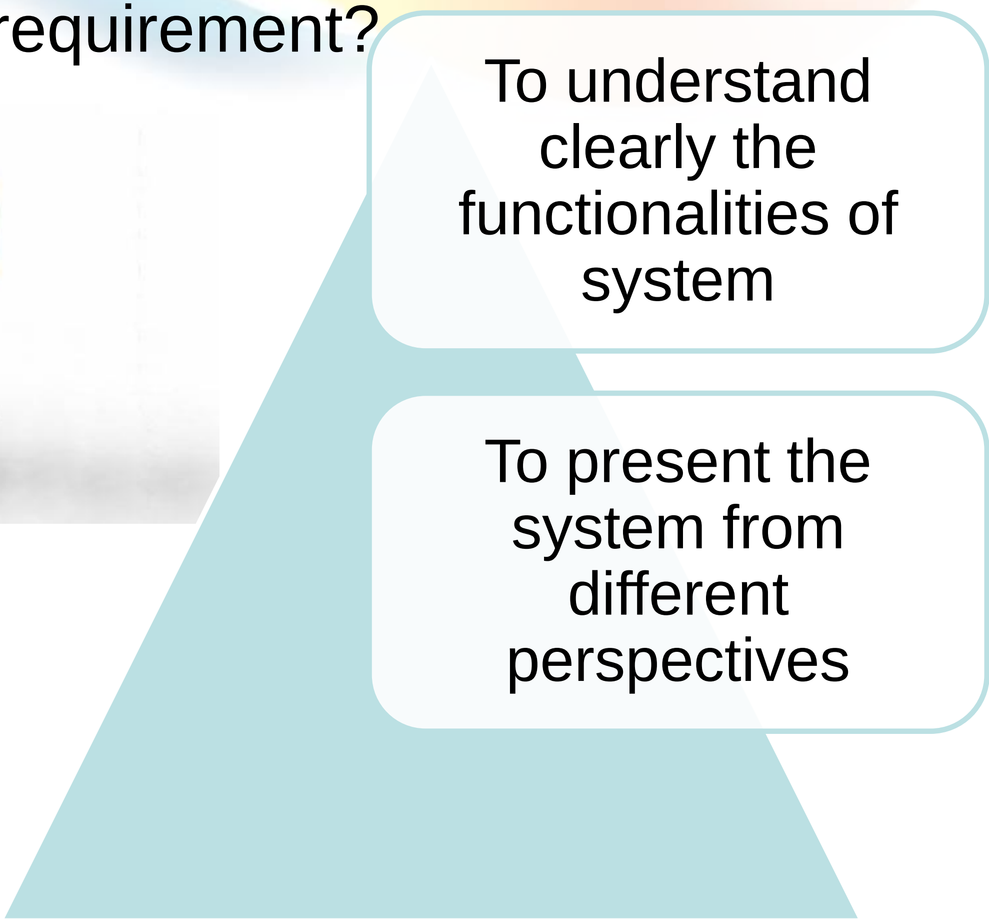
- How to make Q&A effectively?
  - Identify the issue: unclear, get for more information, etc.
  - Check in all documents that customer supplied to make sure your question has not solved;
  - With technical question, check your team /group/company or ask “Google” to solve it before asking out
  - Give the cross-reference clearly, completely
  - Attach sample screen, demo, give your suggestions if any
  - Convert questions to Y/N or multiple-choice types if possible
  - In Q&A, give deadline that you want to receive the answer. It there is no answer until the deadline, what is impact?
  - Take the receiver to re-read the question before sending

# Requirement Clarifying
## Clarifying requirement via Q&A (cont.)

- Q&A focus:
  - Question for idea conveyed by words like: maybe, generally, etc.
  - What is the TBDs - Ask PL to remove all TBDs before handling to you for designing or coding
  - Conflict between requirements. Read the requirement matrix
  - Don’t make assumptions, just ask your PM, PL or BA
- Follow up the Q&A
  - Track the discussion history for easier following up
  - If your question has not been replied or impacts to your task must be report to your PM, BA, or TL immediately
  - Keep in mind your manager/customer are very busy. So it is necessary to remind them about your pending issues daily, weekly. If not, your task will be impacted
- Template on Q&A Management Sheet: 02_SWR_Software Requirement\Student\Assignments\Templates\StudentName - Topic - Q&AList.xls

# Requirement Modeling
# Modeling objectives

- Why model requirement?

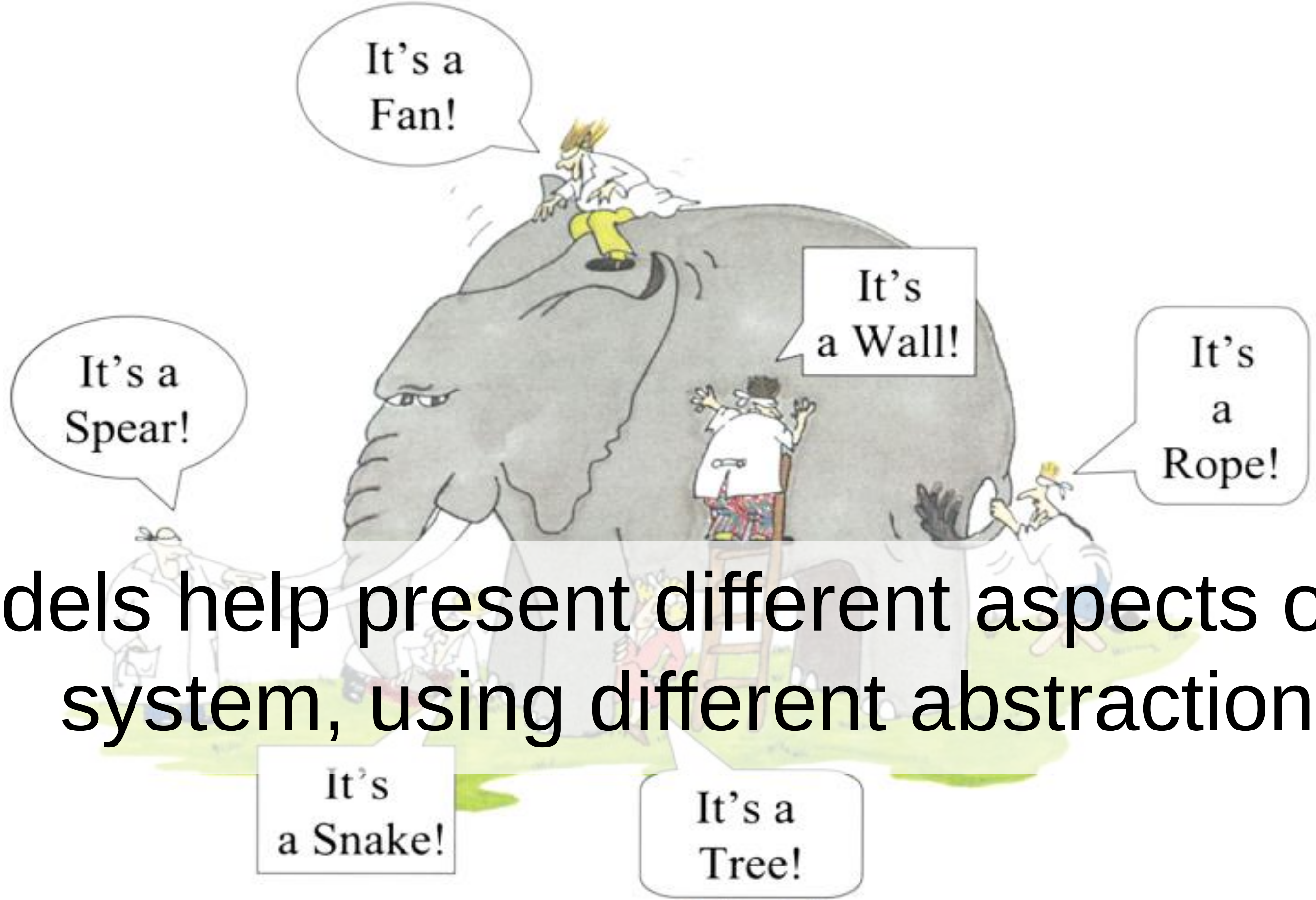


To understand clearly the functionalities of system

To present the system from different perspectives

# Requirement Modeling
# Model different perspectives

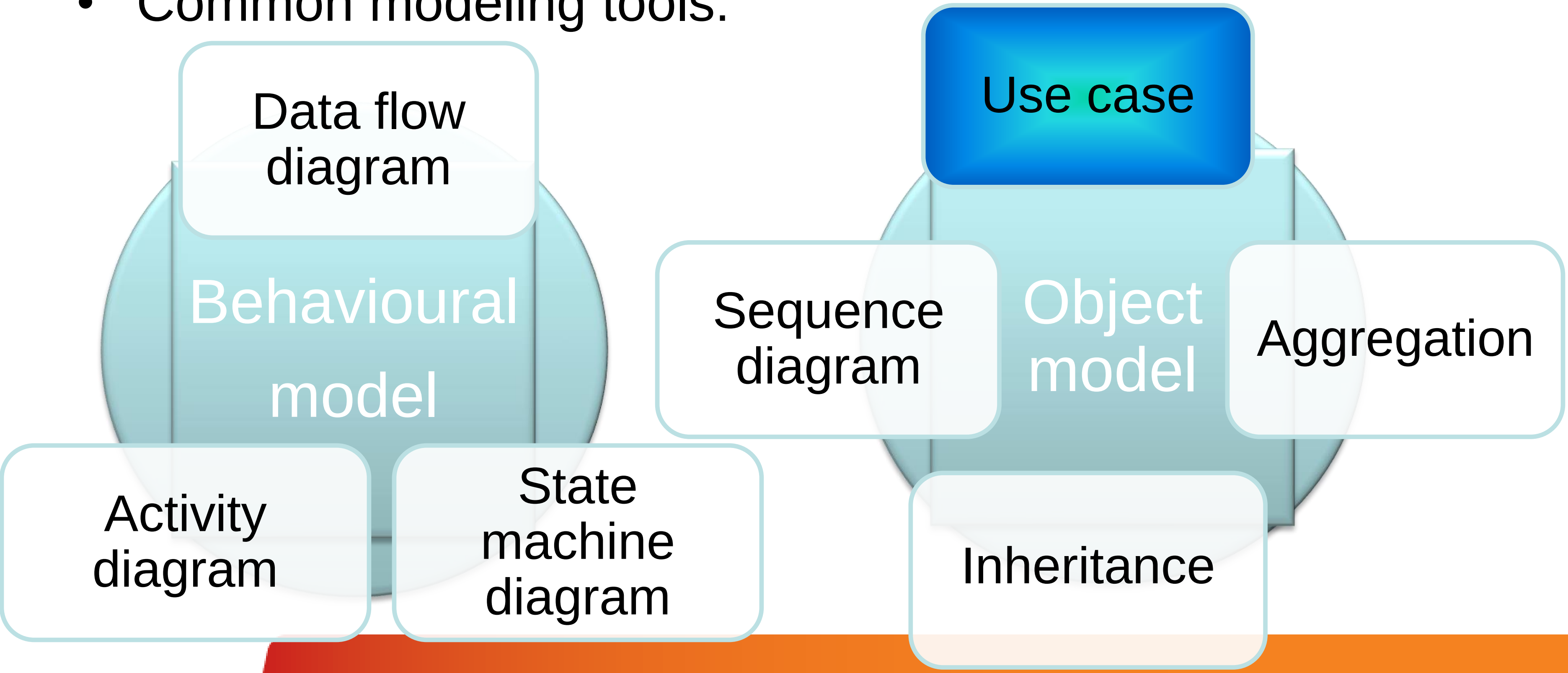
- Model different perspectives



Models help present different aspects of the system, using different abstraction

# Requirement Modeling
# System Modeling Tools

- The system modeling presents an abstraction of the system in software aspects, which helps understanding of the functional requirements in block diagram form, and helps to identify all required software elements & tasks.
- Common modeling tools:



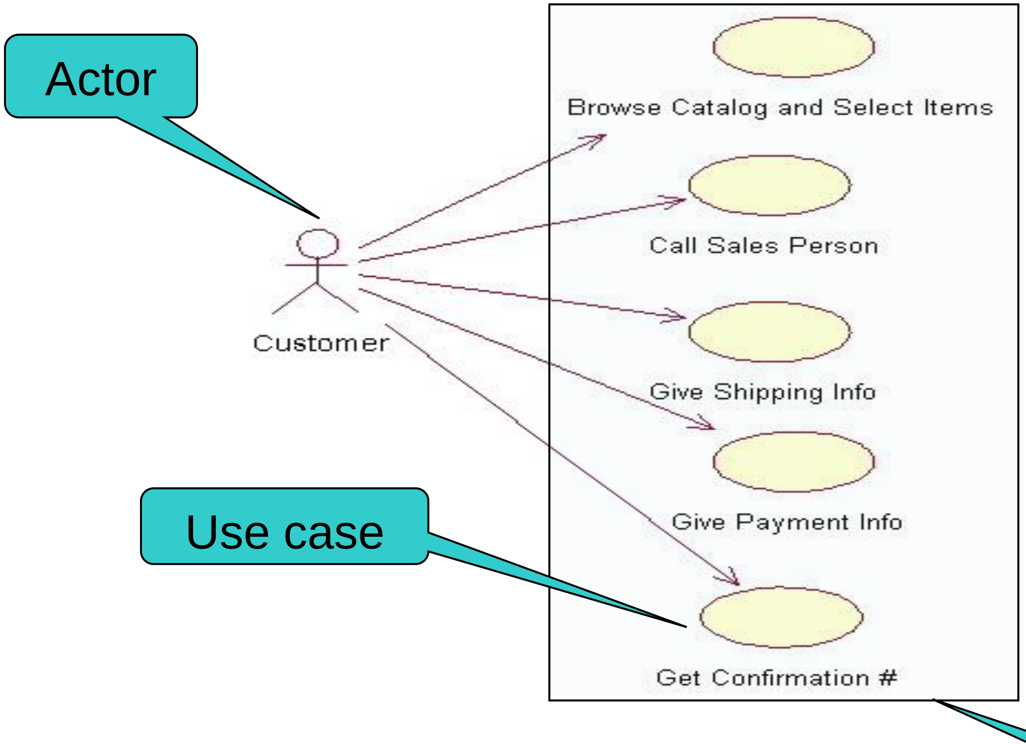
# Modeling Tools - Use Case

- Requirements capture
  - Requirements are reason-for-existence of any software development project
  - Defines and delineates user-requirements
    - Defines the functionality to be provided
    - Identifies the goals to be achieved
  - Must be precisely and completely understood
  - Requirements often changes, thus must be well-documented

# Modeling Tools - Use Case

- Requirement capture with UML
  - Use Case diagram
    - Shows a set of use cases, actors and their relationships
  - Captures problem-domain in terms of:
    - functionality to be provided (Use Cases)
    - the “roles” (Actors) for whom these functions are performed

# Modeling Tools - Use Case
## Use Case Diagram



Online C2C shopping

- overview the usage requirements
- presentations project stakeholders
- "the meat" of the actual requirements

**Actor:**

An actor is a person, organization, or external system that plays a role in one or more interactions with your system

**Use case:**

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse

**System boundary:**

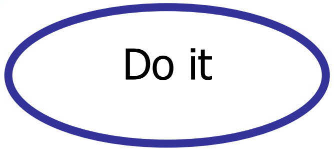
indicates the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not

**System boundary**

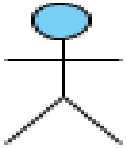
# Modeling Tools - Use Case
# Use Case Diagram - Notations

- Use case diagram: Notations

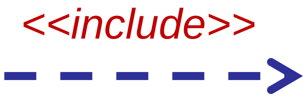
**use case**
A description of a set of sequences of actions that system performs that provides an observable value to an actor



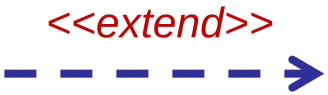
**actor**
The people or systems that provide or receive information from the system; they are among the stakeholders of a system



**include**
Specifies that the source use case explicitly incorporates the behavior of another use case at a location specified by the source



**extend**
Specifies that the target use case extends the behavior of the source.



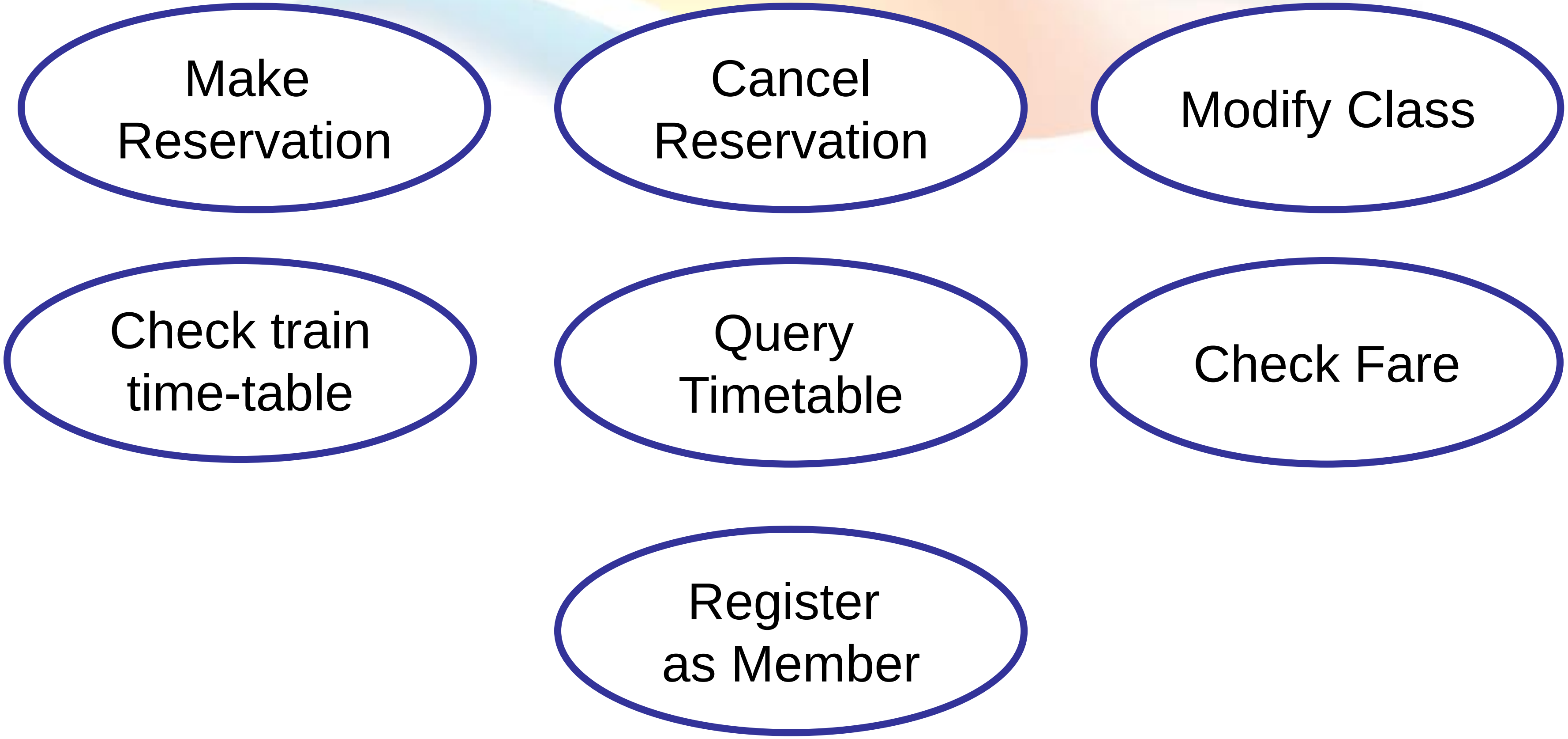
# Modeling Tools - Use Case Example - Use Case 1/2

A company wants to develop a ticketing and reservation system. This must support advance booking of tickets, cancellation of tickets and **change of class** of a ticket. All these are handled by a Reservation Clerk.

The system will also have a Web site where users can **register** themselves and purchase tickets online. They can pay either by using their online banking account or by credit card. **Reservations** made over the internet can only be **cancelled** across the counter.

The system will also have a querying facility that allows users to **check train time-tables**, **fares** and **availability of tickets**.

# Modeling Tools - Use Case Example - Use Case 2/2



Make Reservation

Cancel Reservation

Modify Class

Check train time-table

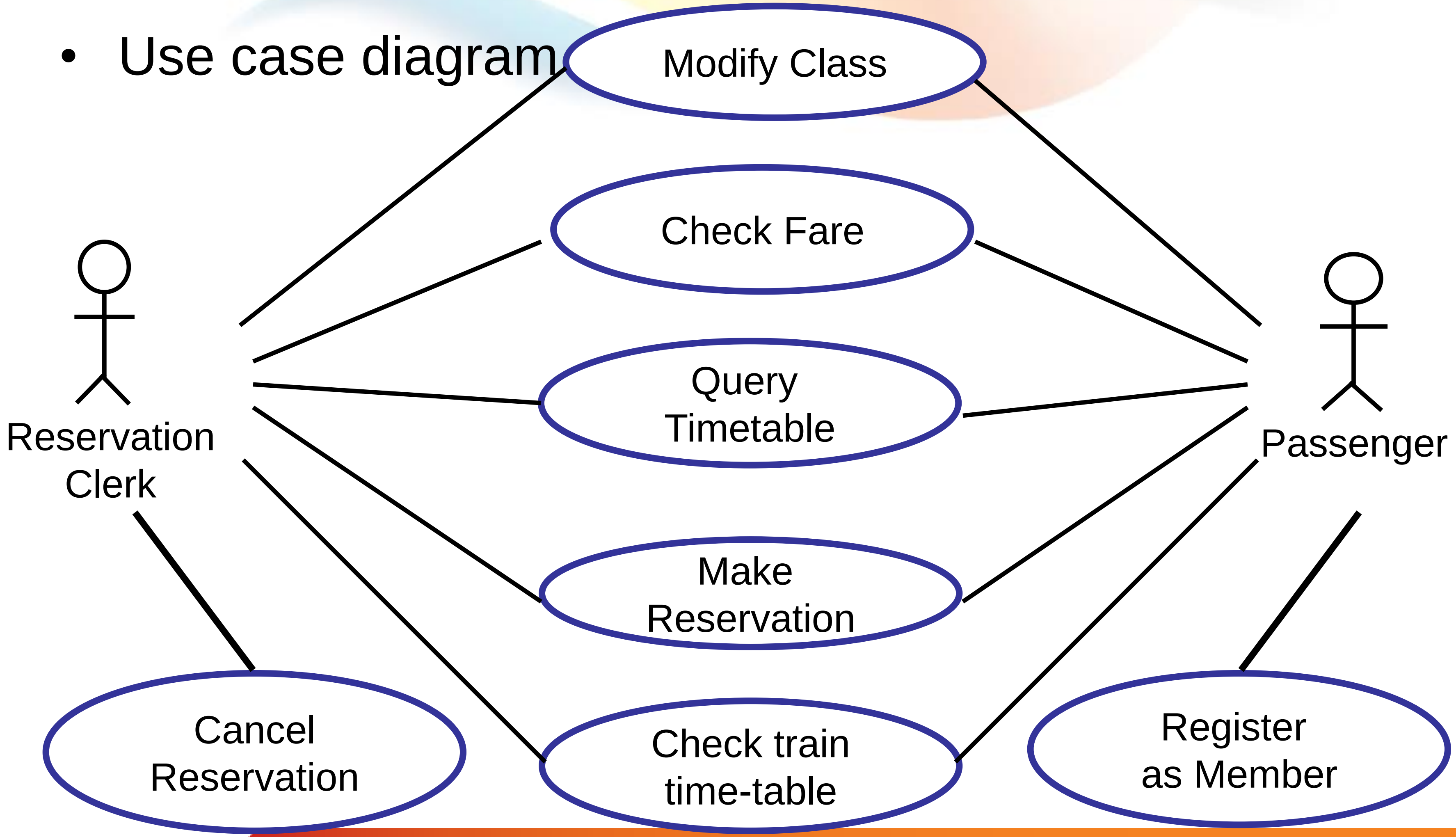
Query Timetable

Check Fare

Register as Member

# Modeling Tools - Use Case
# Example - Use Case Diagram

- Use case diagram



# Modeling Tools - Use Case
# Use Case Specification 1/7

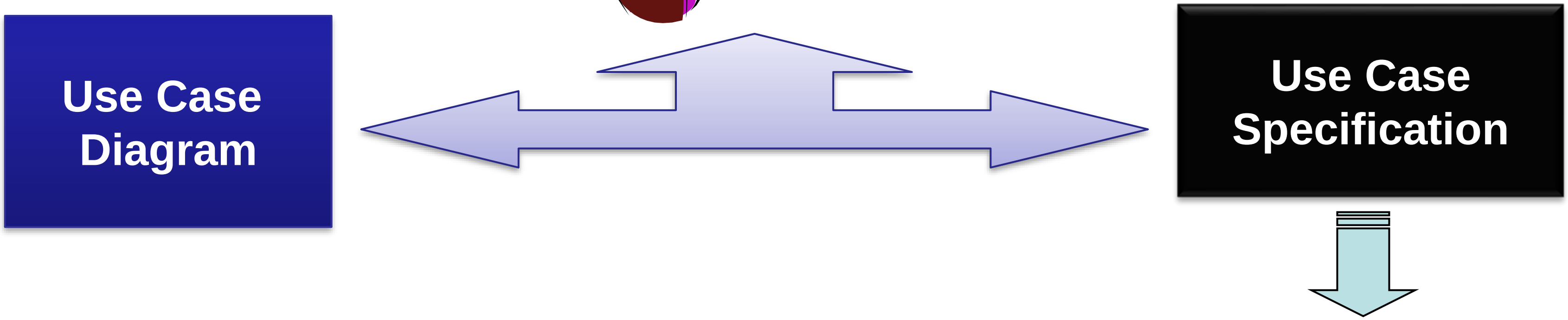
- Text description of use case functionality in the user language and terminology
- No specific UML format
- Describes WHAT and not HOW
- Typically includes:
  - Objectives of the use case
  - How the use case is initiated
  - The flow of events (main flow, alternative flow)
  - How the use case finishes with a value to the actor

  *and more...*

# Modeling Tools - Use Case
# Use Case Specification 2/7

- Use case description serves as a ‘bridge’ between stakeholders of a system and the development team.

**Systems analyst produce use case diagram & use case specification in consultation with end users**



**Use Case Diagram**

**Use Case Specification**

# Modeling Tools - Use Case
# Use Case Specification 3/7

- Flow of events
  - Use Case is an abstraction of behavior (set of sequences)
  - The behavior of the Use Case can be described by a “flow of events” - which spells out in detail what exactly the Use Case does
    - main flow: what happens and in what order when all is well
    - alternate flow(s): what happens and in what order when something goes wrong
    - exception flow: things don’t always go as planned. An exception is an error condition that is important enough to the application to capture

# Modeling Tools - Use Case
# Use Case Specification 4/7

| Key components | Explanation |
|---|---|
| **Name** | *Clear, unique name of the use case (verb, goal-driven)* |
| **Actors** | *Someone or something that interacts with the use case* |
| **Description** | *Brief overview of the use case, describing the main idea* |
| **Goal** | *What the actors achieve with this use case* |
| **Pre-condition** | *State(s) the system can be in before the use case starts* |
| **Trigger** | *Event that causes the use case to be initiated* |
| **Post-condition** | *State(s) the system can be in after the use case finishes* |

# Modeling Tools - Use Case
# Use Case Specification 5/7

| Key components | Explanation |
|---|---|
| **Normal flow** | *Typical (primary) processing path* |
| **Alternative flow** | *Alternative (secondary) processing path* |
| **Exception flow** | *When things go wrong at the system level* |
| **Others** | Business rules, Assumption, Notes, etc. |

# Modeling Tools - Use Case
# Use Case Specification 6/7

- Example

| Make a seat reservation use case | |
|---|---|
| **Name** | *Make reservation* |
| **Actors** | *Passenger* |
| **Description** | *Allows a passenger to book a plane seat for a journey from the Website* |
| **Goal** | *Reserve a seat* |
| **Pre-condition** | *Main Webpage is displayed successfully* |
| **Trigger** | *User clicks on “Reserve seat” button on the main Webpage* |
| **Post-condition** | • *A seat is booked*<br>• *Number of available seats is reduced* |

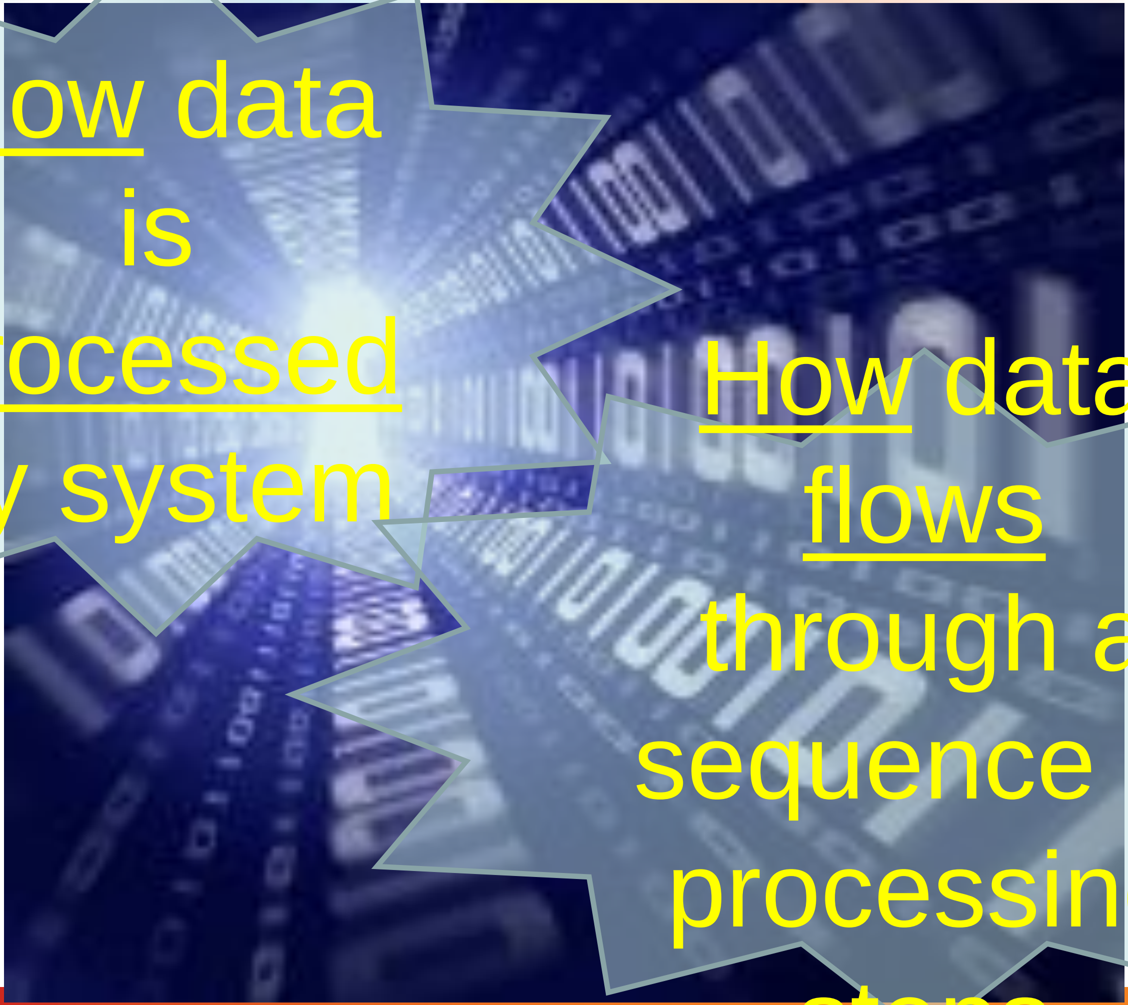
# Modeling Tools - Use Case
# Use Case Specification 7/7

- Example

| **Make a seat reservation use case** | |
|---|---|
| **Normal flow** | *[User log in and reserve a seat successfully]*<br>1. User logs in<br>2. User specifies a flight and travel details<br>3. User specifies passenger details<br>4. User specifies payment details<br>5. User confirms transaction |
| **Alternative flow** | *[When no seat is available on the selected date]*<br>• Show option to select another day<br>• Repeat steps in normal flow |
| **Exception flow** | *[When a payment is failed]*<br>• Notify error with the payment<br>• Give an option to re-enter payment details or other payment method |

# Modeling Tools - DFD

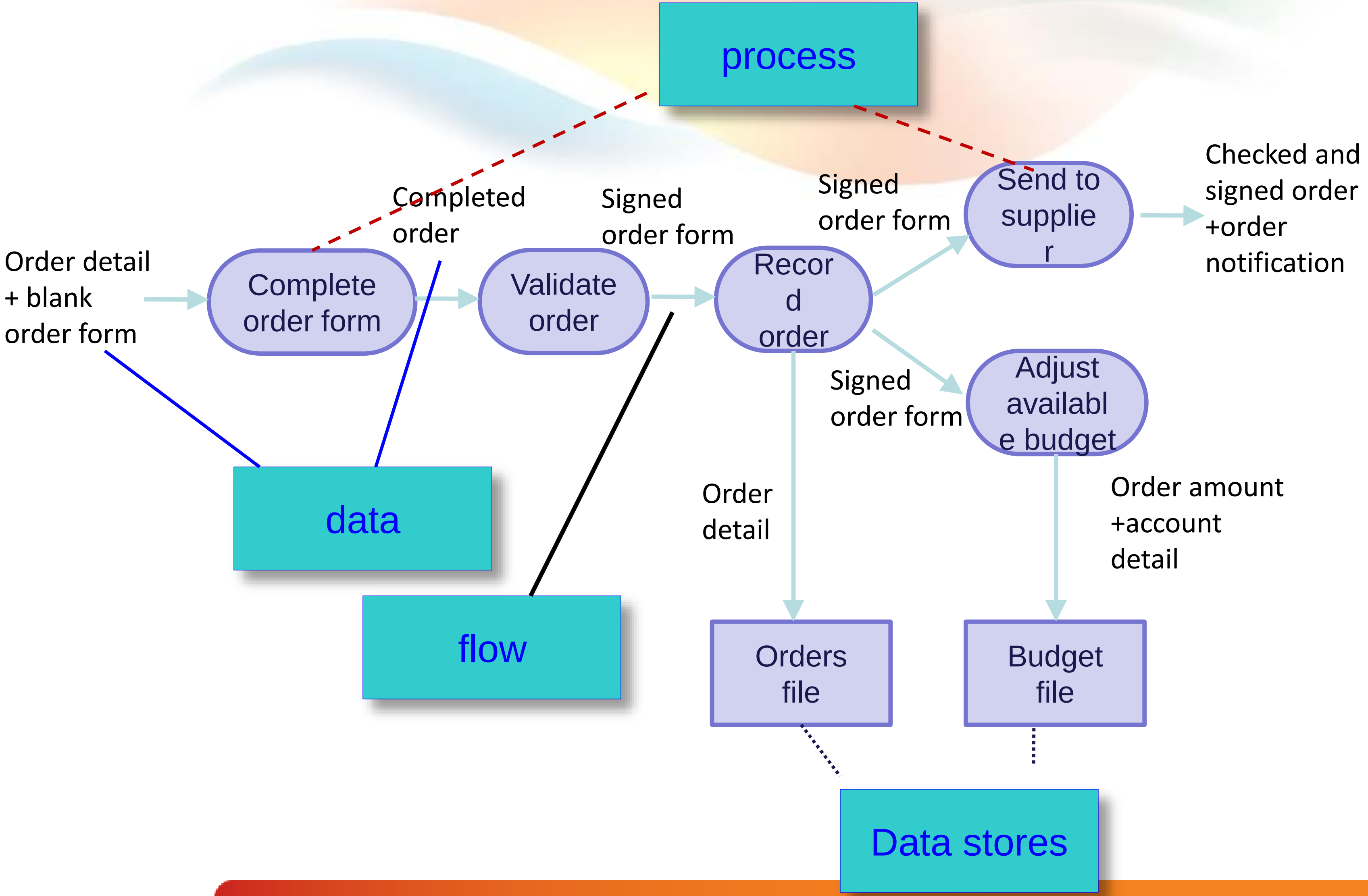
**DFD = Data Flow Diagram**



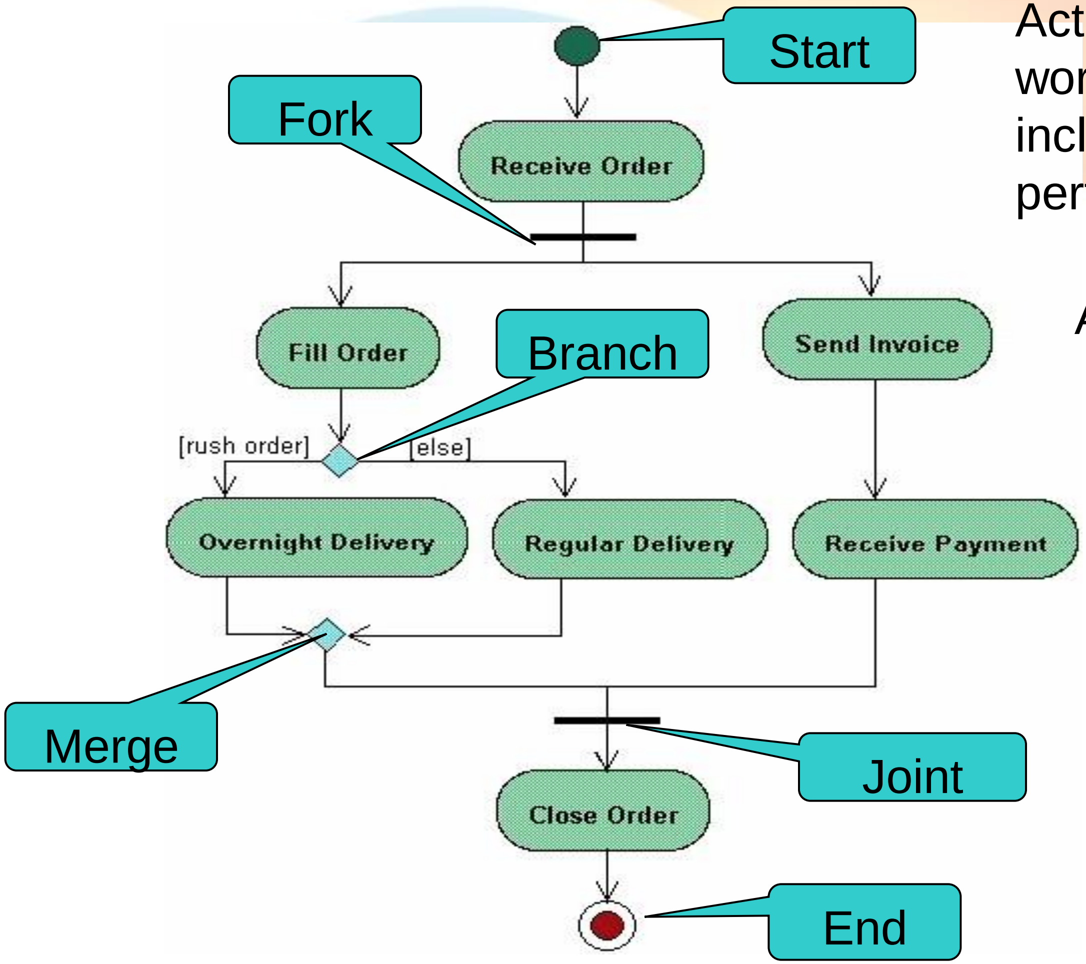
How data is processed by system

How data flows through a sequence of processing steps

# Modeling Tools – DFD Sample DFD

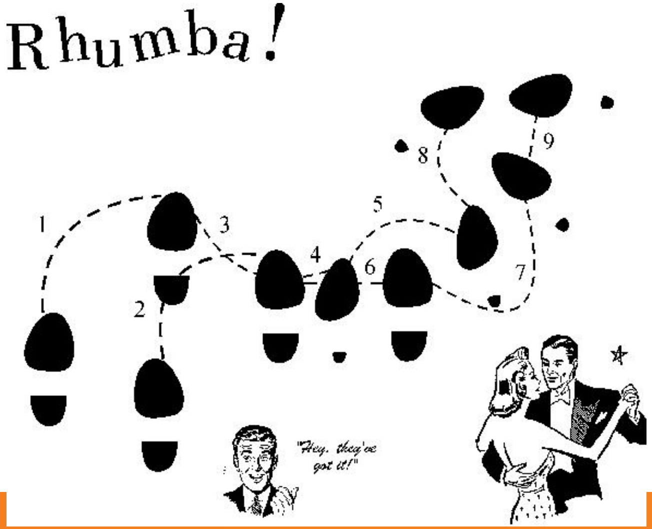


# Modeling Tools - Activities Diagram

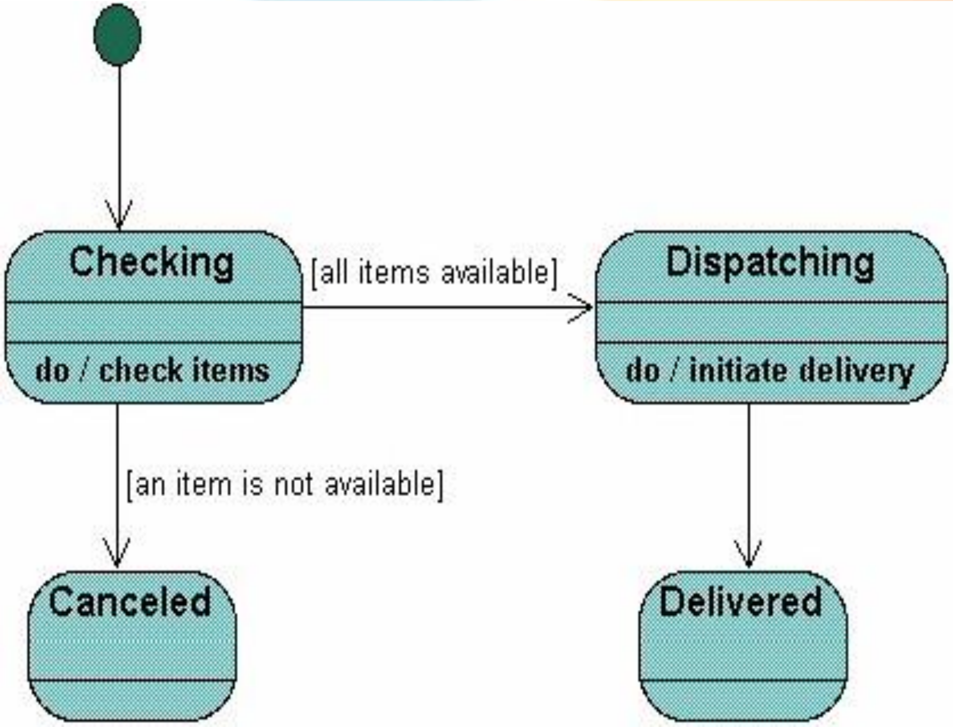


Activity diagrams describe the workflow behaviour of a system including a sequence of activities performed from start to finish

Activities could be performed:
- sequential order
- parallel
- conditional transition



# Modeling Tools - State Machine



A **State Machine diagram** shows the possible states of the object and the transitions that cause a change in state.

?

What is different between activities and Statemachine diagram

# Common Practices, Problems

- Common issues
  - Requirement isn't clear. Don't understand customer mean.
  - Requirement analysis is not documented/centralized recorded -> misunderstanding
  - How to verify the understanding of team members about requirement
- What should we do
  - Maintain SRS
  - Communicate with the customer  = onsite
  - Don’t make assumption. Must confirm with customer about specs more and more about what are still not clear or too general.
  - Should involve all members to investigate requirements from the beginning of the project, not only PL.
  - Conduct meeting to verify the requirement understanding.
  - One meeting to introduce the requirement/design.
  - Another meeting to verified the understanding of team member. Every one have to present their understanding.

# Resources & References

- Resources & References
  - Requirement development & management process
  - Q&A management sheet template
  - SRS review checklist
  - RMS template
  - SRS template

# Q & A