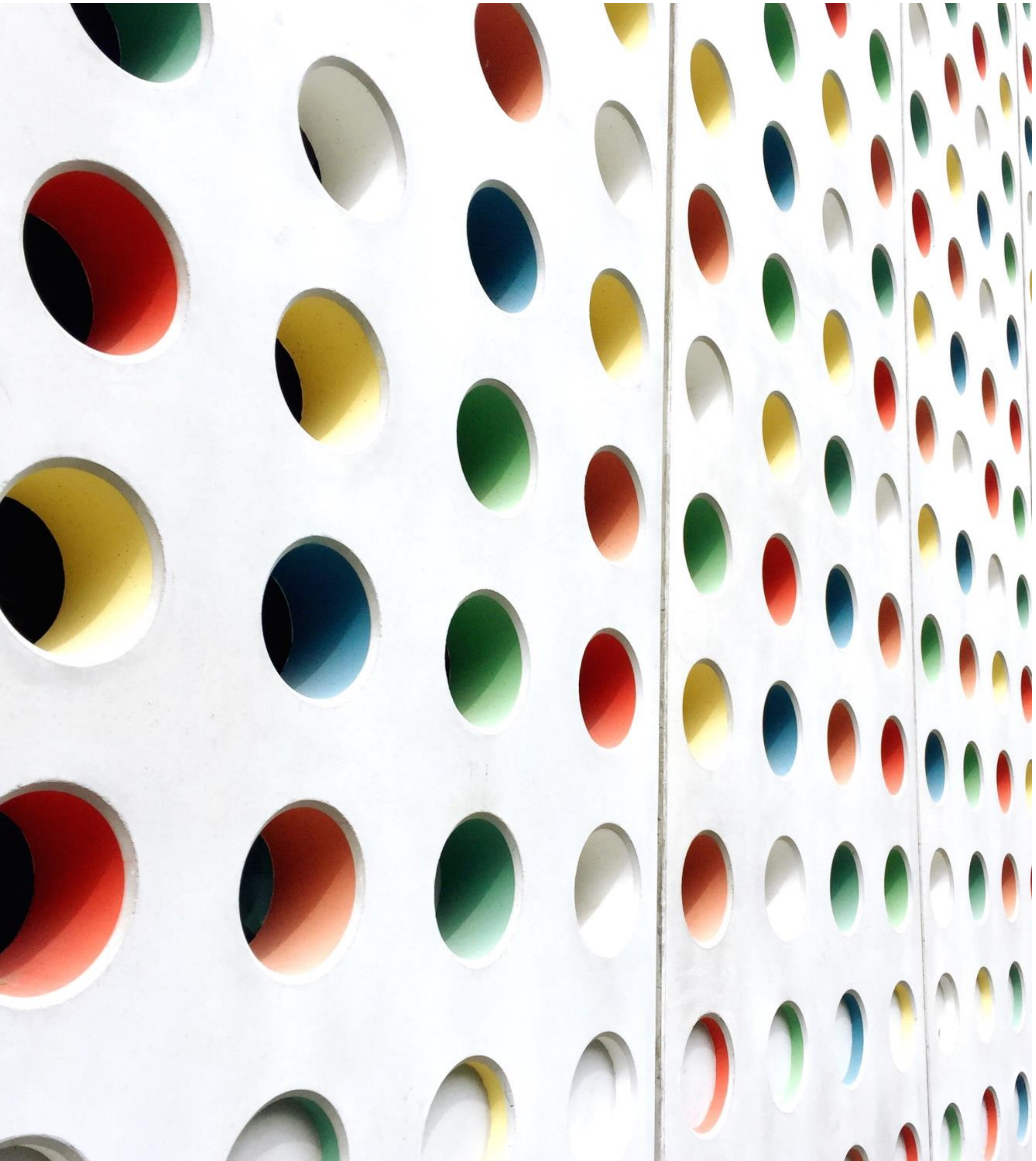


Analysis Model & Requirement Allocation

Systems Engineering Fundamentals

Requirements Analysis

Requirements Analysis Techniques

- Context Analysis
- States and Modes Analysis
- Requirements Parsing
- Requirements Template
- Operational Concept Analysis
- Requirements Decomposition
- Use Case Analysis
- ViewPoints Oriented Requirements Development

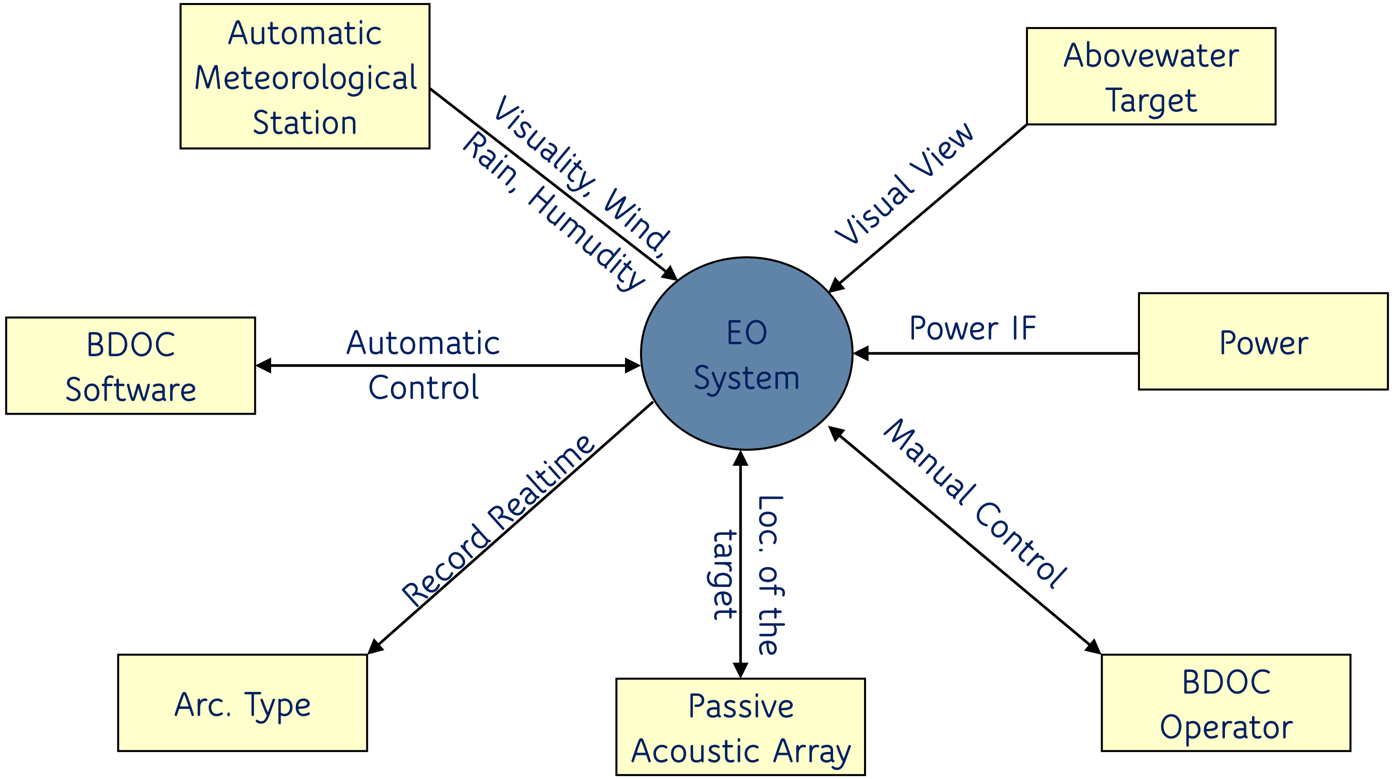
Context Analysis

- Capturing all specified interfacing systems and (life cycle) system inputs and outputs to provide a progressively developed point of reference for identification of missing external interface requirements and inputs/outputs.

Purpose of The Context Analysis

- It provides a much better understanding of the requirements, and of the nature and extent of requirements issues.
- Work through source documents, identifying each reference to an external system or particular element of the environment which is required to interface or interoperate with the system, or to an item which is specified to be input to or output from the system.

Conduct Context Analysis



Name the interfacing entities and related external interfaces.

Requirements Analysis Techniques

- Context Analysis
- States and Modes Analysis
- Requirements Parsing
- Requirements Template
- Operational Concept Analysis
- Requirements Decomposition
- Use Case Analysis
- ViewPoints Oriented Requirements Development

States and Modes Analysis

States and Modes Analysis is performed to gain and effectively communicate an understanding of groupings, at the highest level of abstraction, of time-variant alternative characteristics required of the system.

Conduct States and Modes Analysis

For each state/mode prepare dictionary description of that state/mode.

- State/Mode Dictionary Entries Example:
- **The Limited Capability State** of the Abovewater Surveillaince and Detection System in which one of the EO sensor is off due to malfunction or maintenance. In this state other EO sensors will try to cover the surveillance region of the failed sensor.

- **Automatic Scanning Mode** of the Abovewater Surveillance and Detection System is that mode EO sensors scan their predefined sectors for target detection.

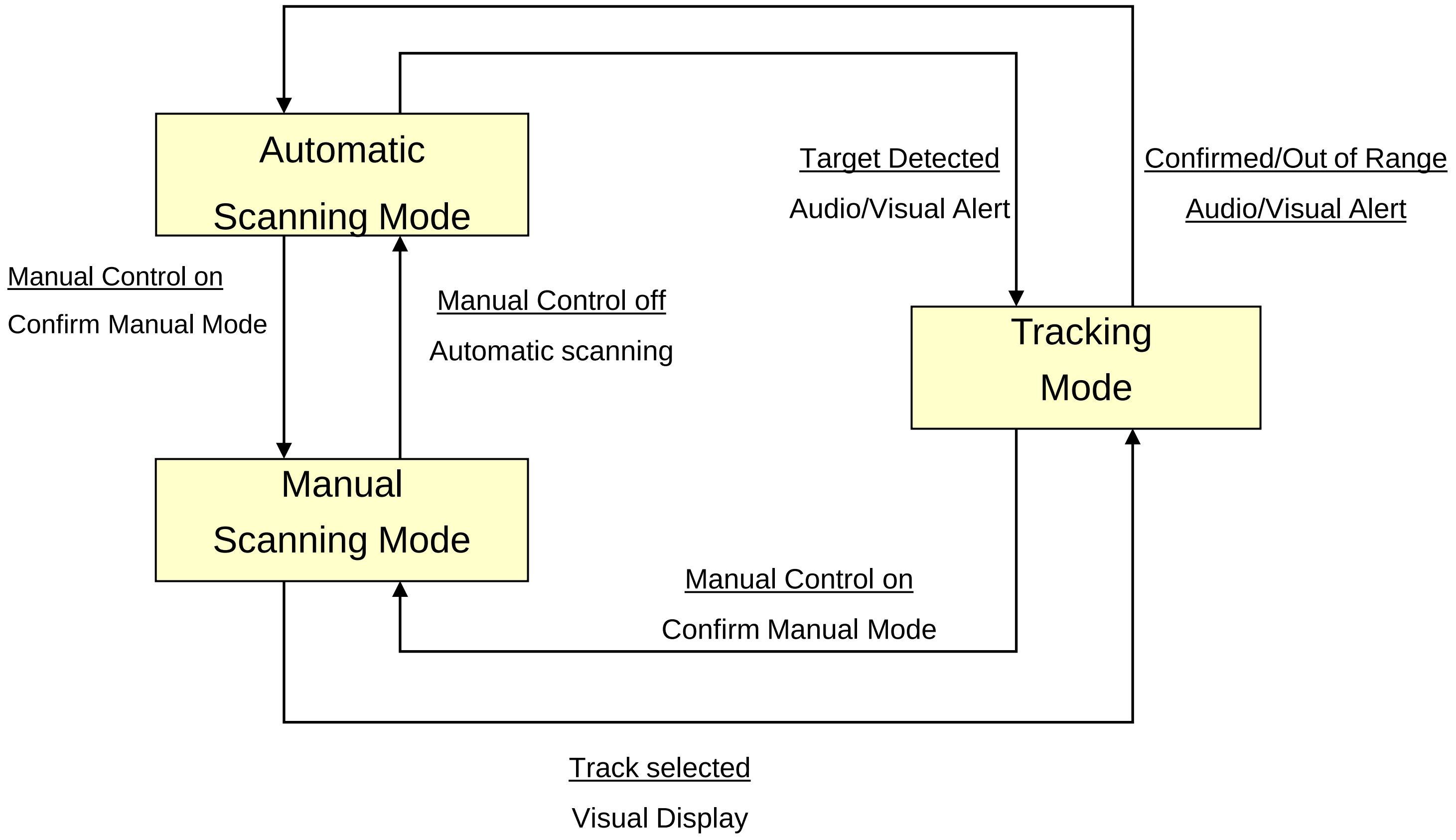
Conduct States and Modes Analysis

Enter the identified states and modes into a state/ mode table. Show “mutual exclusivity” in the table

State/Mode Table Example:

State/Mode	Limited Capability State	Full Capability State	Automatic Scanning Mode	Manual Scanning Mode	Tracking Mode
Limited Capability State		M		M	M
Full Capability State	M				
Automatic Scanning Mode				M	M
Manual Scanning Mode	M		M		M
Tracking Mode	M		M	M	

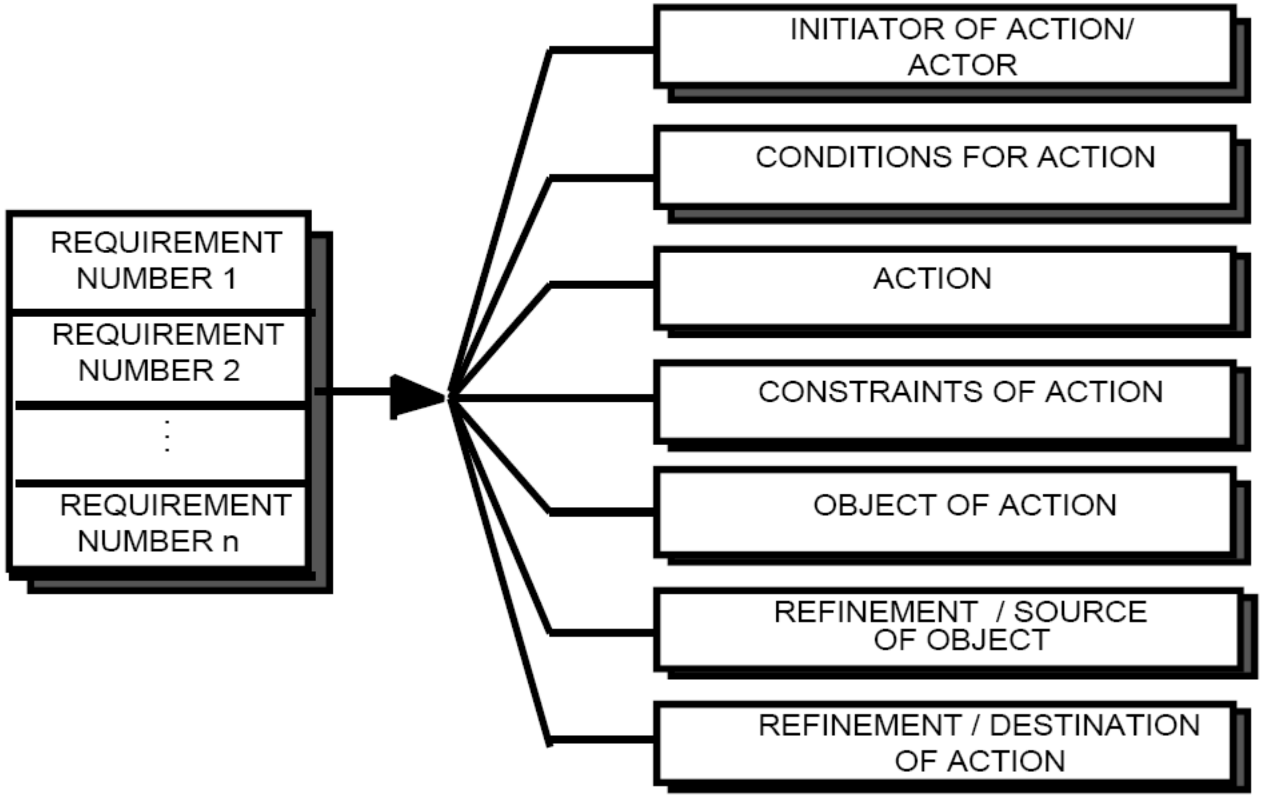
Conduct States and Modes Analysis



Requirements Analysis Techniques

- Context Analysis
- States and Modes Analysis
- Requirements Parsing
- Requirements Template
- Operational Concept Analysis
- Requirements Decomposition
- Use Case Analysis
- ViewPoints Oriented Requirements Development

Requirements Parsing



Requirements Parsing is a text analysis technique for identification of error, incompleteness, inconsistency, lack of clarity, ambiguity, unverifiability and infeasibility in textually stated requirements.

Elements of Parsing Template

- ***Actor/Iniator of Action.*** This is the subject of the sentence - the thing being specified. Examples are: "the system", "the interface", "the function".
- ***Action.*** This is a verb the action to be taken by the actor (subject). Examples are "shall calculate", shall display", "shall fly".
- ***Object of Action.*** This is a noun, and is the thing acted upon by the actor. Examples are: "the message", "the input signal".
- ***Conditions of Action.*** This defines the conditions under which the action takes place, for example "upon receipt of a message", "in high resolution mode", "within 10 minutes of power-on".
- ***Constraints of Action.*** This qualifies the action, for example "at a resolution of 4000 x 1000 pixels", "within limits imposed by vehicle speed".
- ***Refinement/Source of Object.*** These qualify the object, for example (refinement): "of flash priority", for example (source): "from DISCON".
- ***Refinement/Destination of Action.*** These further qualify the action, and may be additional to Constraints of Action. Examples are "within 10ms", "to DISCON".
- ***Other.*** This element collects non-requirements material.

Requirement Parsing

The system, shall switch the message within 10 milliseconds of receipt, upon receipt of a message for messages in ACP128 format having a valid routing indicator, from the message input port, to a message output port, corresponding to the routing indicator in the message.

Actor	***The system,***
Condition of Action	***upon receipt of a message***
Action	***shall switch***
Object of Action	***the message***
Constraints of Action	***within 10 milliseconds of receipt,***
Refinement of Object	***for messages in ACP128 format having a valid routing indicator,***
Source of Object	***from the message input port,***
Destination of Action	***to a message output port,***
(Further) Refinement of Action	***corresponding to the routing indicator in the message.***

Conduct Requirements Parsing

For the given Zone, a main Switch, which is identical to a trunk node switch, shall be given two (2) independent links to at least two (2) other nodes in the network.

Actor	***A main Switch,***
Condition of Action	***For the given zone,***
Action	***shall be given***
Object of Action	***Two (2) independent links***
Constraints of Action	
Refinement of Object	***to at least two (2) other nodes in the network***
Source of Object	
Destination of Action	
(Further) Refinement of Action	***which is identical to a trunk node switch,***

Conduct Requirements Parsing

The operator shall be able to read a 10 cm high name and other distinguishing markings at a range of at least 0.25 nm and a 40 cm high name at a range of 1 nm during the day under high visibility conditions and with a black-white contrast.

Element	Text	Remark
Actor	The operator	
Action	shall be able to read	
Object of Action	a 10 cm high name and other distinguishing markings / a 40 cm high name	Two objects?
Conditions for Action	at a range of at least 0.25 nm /at a range of 1 nm	Two objects?
Constraints of Action	during the day under high visibility conditions	High visibility?
Refinement/ Source of Object		
Refinement/ Destination of Action	and with a black-white contrast.	Contrast scale?

Conduct Requirements Parsing

- Calculate Individual Requirement Quality (IRQ)
  - Determine which of the possible seven elements of the structure are applicable and assign a value of 1 to each applicable element (most requirements have 5-7 applicable elements);
  - Assess each element of the parsed requirement against the quality factor criteria, and score each applicable element as 1 (satisfactory) or 0 (unsatisfactory).
  - Calculate the metric by dividing the sum of the element scores into the sum of the applicable element values.

Conduct Requirements Parsing

Example IRQ:

Element	*Applicability*	*Score*
Actor	*1*	*1*
Action	*1*	*1*
Object of Action	*1*	*0*
Conditions for Action	*1*	*1*
Constraints of Action	*1*	*0*
Refinement/ Source of Object	*0*	*0*
Refinement/ Destination of Action	*1*	*0*
TOTAL	*6*	*3*

- OmmissionRatio =TotalApplicability-TotalScore = 3
- IRQ = TotalScore / TotalApplicability =3 / 6 = 0.5

Conduct Requirements Parsing

Score Indivudual Quality Factors (IQF)

Quality Metrics	***Metric Name***	***Metric Value***
Correctness	***IQF1***	***1***
Completeness	***IQF2***	***1***
Consistency	***IQF3***	***1***
Clarity	***IQF4***	***0***
Non-ambiguity	***IQF5***	***1***
Singularity	***IQF6***	***0***
Verifiability	***IQF7***	***0***
Feasibility	***IQF8***	***1***
Functional Orientation	***IQF9***	***1***

Conduct Requirements Parsing

- Calculate Aggregate Requirements Quality and Quality Factors from individual requirements for requirements groups

  - RQ =∑IRQ/n   RQ means Requirements quality
  - QF1 = ∑IQF1/n QF means Requirements Quality Factor
  - IQF2 = ∑IQF2/n - ∑OmmisionRatio/n
  - IQFx = ∑IQFx/n x takes a value between 3 and 9

Conduct Requirements Parsing

Metrics	QL1	QL2	QL3	QL4
RQ	0.01-0.3	0.3-0.7	0.95-0.99	0.99+
QF1-Correctness	0.9	0.98	0.99	0.99+
QF2-Completeness	-0.5	0.4	0.95	0.99+
QF3-Consistency	0.9	0.97	0.99	0.99+
QF4-Clarity	0.9	0.97	0.99	0.99+
QF5-Non-Ambiguity	0.3	0.7	0.9	0.98+
QF6-Singularity	0.1	0.3	0.99+	1
QF7-Verifiability	0.1	0.7	0.99	0.99+
QF8-Feasibility	0.95	0.99	0.99+	0.99+
QF9-Functional Ori.	0.9	0.98	0.99	0.99+

- Quality Level (QL) of Requirements Set
  - ***QL1:*** Very poor set of requirements, requiring substantial development
  - ***QL2:*** Fair set of requirements, may just be suitable for purposes of solicitation, depending on the SOW and type of contract envisaged
  - ***QL3:*** Requirements at SRR suitable for carrying forward into development
  - ***QL4:*** Requirements suitable for establishment of the Functional Baseline

Conduct Requirements Parsing

- Try to solve deficiencies (to increase quality) with below alternatives;
    - Refine the requirement
    - Derive new requirements
    - Split into child requirements
    - putting a note on decision table as “To Be Resolved” to resolve with customer.(Future resolution)

Requirements Analysis Techniques

- Context Analysis
- States and Modes Analysis
- Requirements Parsing
- Requirements Template
- Functional Analysis
- Operational Concept Analysis
- Requirements Decomposition
- Use Case Analysis
- ViewPoints Oriented Requirements Development

Requirements Template

Requirements Templates standardizes format, simplifies writing procedure, conforms to properties of good requirements, consolidates functionality, interfaces, performance and some specialty requirements such as security.

Capture requirements according to template and try to resolve missing parts by interviewing with stakeholders.

Conducting Requirements Template

- Define requirements template structure to capture need.

The <system_name> shall produce <output>

for use by <destination>

if <trigger>

using <inputs>

where <amplifying information>

Conducting Requirements Template

Define templates variables

system _name - the system/subsystem which produces the output

output – the product that is generated by the system/subsystem

destination(s) –the human user, or an external system that uses the output identified above

trigger –the condition that causes the system to produce the output

input(s) –the information entering the system or subsystem needed to create the output

amplifying information –additional data that clarifies the clauses of the requirement

Example Captured Requirement

The OIS shall produce a missile mission planning display

for use by the OIS user

if requested by the user

using user entered information

where the display includes

available surface launched missile missions,

ground targets,

maritime targets,

missile count and type for user selected missile mission

Requirements Analysis Techniques

- Context Analysis
- States and Modes Analysis
- Requirements Parsing
- Requirements Template
- Operational Concept Analysis
- Requirements Decomposition
- Use Case Analysis
- ViewPoints Oriented Requirements Development

Concept of Operations

- Operational Concept Document (OCD) is produced early in requirements development process
  - to describe what the system will do (not how it will do) and why (system rationale)
  - to define critical, top-level performance requirements or objectives.
- OCD should contain preliminary functional block diagram of the system with only the top-level functional decomposition.
- No attempt is made at this stage to define a complete operational concept.
- OCD is essentially a functional concept definition and rationale from the user and customer perspective.

Objective of OCD Analysis

- To provide traceability between operational needs and the written source requirements captured.
- To establish a basis for requirements to support the system over its life, such as personnel requirements, support requirements, etc.
- To establish a basis for test planning, system-level test requirements, and any requirements for environmental simulators.
- To generate operational analysis models to test the validity of external interfaces between the system and its environment, including interactions with external systems.
- To provide the basis for computation of system capacity, behavior under/overload, and mission effectiveness calculations.
- To validate requirements at all levels and to discover implicit requirements overlooked in the source documents.

How to produce OCD

- Deduce a set of statements describing the higher-level, mission-oriented system objectives (using user’s statement of need and other sources).
- Review the system objectives with end users and operational personnel.
- Define the boundaries of the operational models.
- For each model, generate a context diagram to represent the model boundary.
- Add concurrent functions to the context diagram, which are performed by the sections of external systems that send input stimuli to the system or receive outputs from the system.
- Identify all of the possible types of observable input and output events that can occur between the system and its in interacting external systems.
- Define a system interface between the system and the environment or external system.

How to produce OCD, cont

- For each class of interaction between a part of the system and an external system, create a functional flow diagram to model the sequence of interactions as triggered by the stimuli events generated by the external systems.
- Review the functional flow diagrams with end users and operational personnel.
- Develop timelines, approved by end users, to supplement the source requirements.

OCD Topics

1. Scope
    - 1.1 Identification
    - 1.2 System Overview
    - 1.3 Document Overview
2. Reference Documents
3. Current System or Situation
    - 3.1 Background, Objectives, and Scope
    - 3.2 Operational Policies and Constraints
    - 3.3 Description of Current System or Situation
    - 3.4 User or Involved Personnel
    - 3.5 Support Concept
4. Justification for and Nature of Changes
    - 4.1 Justification for Change
    - 4.2 Description of Needed Changes
    - 4.3 Priorities Among the Changes
    - 4.4 Changes Considered but Not Included
    - 4.4 Assumptions and Constraints
5. Concept for a New or Modified System
    - 5.1 Background, Objectives, and Scope
    - 5.2 Operational Policies and Constraints
    - 5.3 Description of the New or Modified System
    - 5.4 Users/Affected Personnel
    - 5.5 Support Concept
6. Operational Scenarios
7. Summary of Impacts
    - 7.1 Operational Impacts
    - 7.2 Organizational Impacts
    - 7.3 Impacts During Development
8. Analysis of the Proposed System
    - 8.1 Summary of Advantages
    - 8.2 Summary of Disadvantages/Limitations
    - 8.3 Alternatives and Trade-offs Considered
9. Notes

Requirements Analysis Techniques

- Context Analysis
- States and Modes Analysis
- Requirements Parsing
- Requirements Template
- Functional Analysis
- Operational Concept Analysis
- Requirements Decomposition
- Use Case Analysis

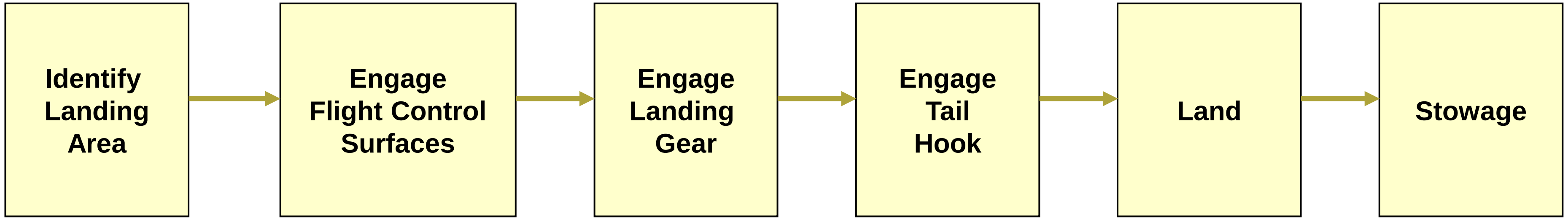
Requirements Decomposition

- Breaking down complex requirements into single, simple requirement statements
- Decomposed requirement statement may interpret or quantify the spec language
- Decomposition does not change the contractual requirement or specification document

Conducting Requirements Decomposition (1)

# 22 – The Aircraft(A/C) shall have a precision landing capability for both Carrier Vessel(CV) and land based operations.

Example Functional Flow:



Requirements Analysis Techniques

- Context Analysis
- States and Modes Analysis
- Requirements Parsing
- Requirements Template
- Functional Analysis
- Operational Concept Analysis
- Requirements Decomposition
- Use Case Analysis
- ViewPoints Oriented Requirements Development

Use Case

- Describes WHAT the system (as a “Black Box”) does from a user’s (actor) perspective
- The Use Case Model is NOT an inherently object oriented modeling technique

Purpose of The Use Case

- Captures operational requirements from user’s perspective
- Gives a clear and consistent description of what the system should do
- A basis for performing system tests
- Provides the ability to trace functional requirements into actual classes and operations in the system

UML Use Case Diagrams

A Use Case model is described in UML (Unified Modeling Language) as one or more Use Case Diagrams (UCDs)

A UCD has 4 major elements:

- **The *system*** described
- **The *actors*** that the system interacts with
- **The *use-cases***, or ***services***, that the system knows how to perform
- **The *relationships*** between the above elements

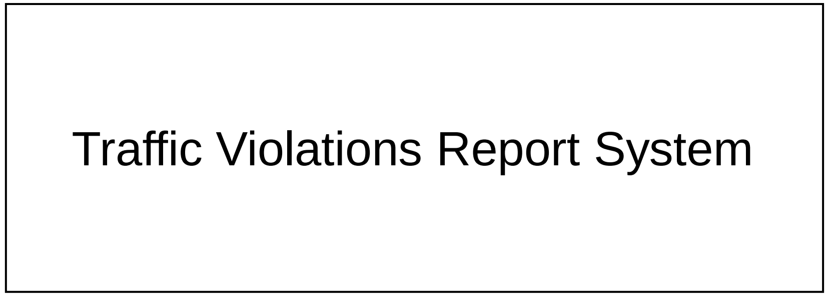
System Boundary in Use Case (1)

- As part of use-case modeling, the **boundaries of the system** developed must be defined
- Defining the boundaries of the system is not trivial
  - Which tasks are automated and which are manual?
  - Which tasks are performed by other systems?
  - The entire solution that we supply should be included in the system boundaries
  - Incremental releases

System Boundary in Use Case

A system in a UCD is represented as a box

The name of the system appears above or inside the box



Traffic Violations Report System

Actor

Someone or something that interacts with the system (exchanges information with the system)

An actor represents a role played with respect to the system, not an individual user of the system

Example:

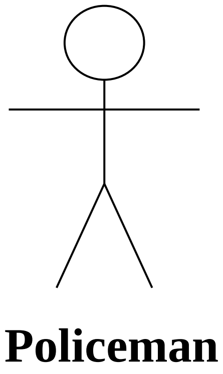
Policeman – Enters data

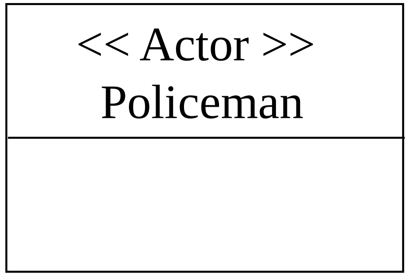
Supervisor – Allowed to modify/erase data

Manager – Allowed to view statistics.   (A single user may play more than one role)

Actor (2)

- Actors have **goals**:
  - Add a Traffic Violation
  - Lookup a Traffic Violation
- Actors don’t need to be human
  - May be an external system that interfaces with the developed system
- An actor has a name that reflects its role





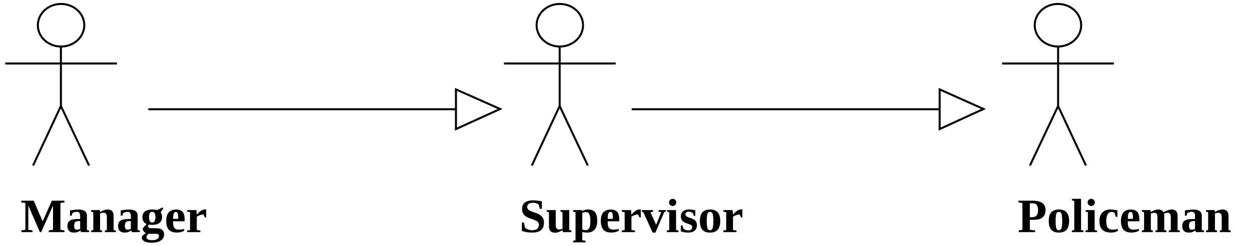
Relationships between Actors

When several actors as part of their roles, also play a more generalized role, it is described as generalization

The behavior of the general role is described in an actor super-class

The specialized actors inherit the behavior of the super-class and extend it in some way

Relationships between actors are not always necessary



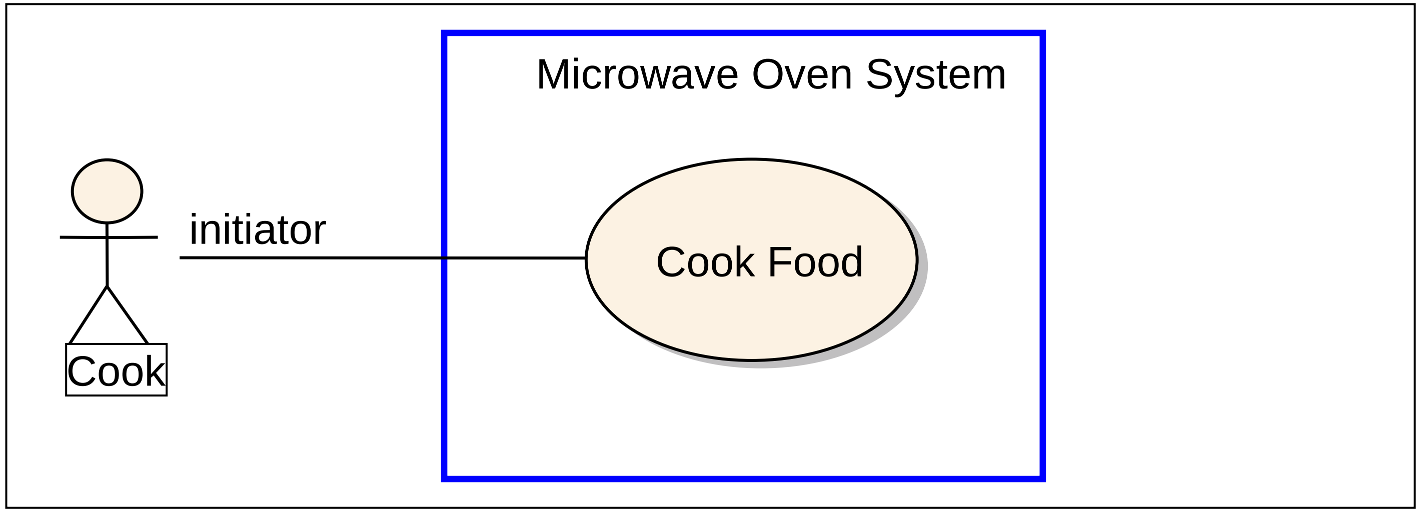
Use Case

- Represent a complete behavior as perceived by an actor
- A use case satisfies an actor’s goal
- Always initiated by an actor
- A use case is complete
  - Don’t divide a use case into smaller use cases that implement each other (functional decomposition)

Conduct Use Case (1)

Draw use case diagram that shows actors and interactions of the actors with the system

Example: (Microwave Oven System)



Conduct Use Case (2)

Define a use case for each use case of the use case diagram

Example:

Name: Cook Food

Brief description: This use case describes the user interaction and operation of a microwave oven. Cook invokes this use case in order to cook food in the microwave.

Added value: Food is cooked.

Scope: A microwave oven

Primary actor: Cook

Supporting actors: Timer

Preconditions: The microwave is waiting to be used.

Conduct Use Case (3)

Define main success scenario Example:

1. Cook opens the microwave oven door, puts food into the oven, and then closes the door.
2. Cook specifies a cooking time.
3. System displays entered cooking time to Cook.
4. Cook starts the system.
5. System cooks the food, and continuously displays the remaining cooking time.
6. Timer indicates that the specified cooking time has been reached, and notifies the system.
7. System stops cooking and displays a visual and audio signal to indicate that cooking is completed.
8. Cook opens the door, removes food, then closes the door.
9. System resets the display.

Conduct Use Case (4)

Define alternate flows

Example:

1a. If Cook does not close the door before starting the system (step 4), the system will not start.

4a. If Cook starts the system without placing food inside the system, the system will not start.

4b. If Cook enters a cooking time equal to zero, the system will not start.

5a. If Cook opens the door during cooking, the system will stop cooking. Cook can either close the door and restart the system (continue at step 5), or Cook can cancel cooking.

5b. Cook cancels cooking. The system stops cooking. Cook may start the system and resume at step 5. Alternatively, Cook may reset the microwave to its initial state (cancel timer and clear displays).

Conduct Use Case (5a)

Capture functional requirements from the steps of the main and the alternative flows

Example:

Req. F1: The system shall provide a mechanism for Cook to enter a cooking time. [From step 2]

Req F2: The system shall be capable of displaying the cooking time entered by Cook. [From step 3]

Req F3: The system shall cook food using microwave radiation. [From step 5]

Req F4: The system shall be capable of calculating and displaying the remaining cooking time [From step 5]

Req F5: The system shall interface with a timer mechanism such that the system is stopped when the timer elapses. [From step 6]

Conduct Use Case (5b)

Capture functional requirements from the steps of the main and the alternative flows

Example (Continued):

Req F7: The system shall emit an audible signal when the timer has elapsed. [From step 7]

Req F8: The system shall visually indicate when the timer has elapsed. [From step 7]

Req F9: The system shall be capable of determining whether the oven door has been closed. [From step 1a]

Req F10: The system shall not start, if the system detects that the door is open. [From step 1a]

Req F11: The system shall be capable of determining if food has been placed in the oven. [From step 4a]

Conduct Use Case (5c)

Capture functional requirements from the steps of the main and the alternative flows

Example (Continued):

Req F12: The system shall not start, if the system detects that no food has been placed in the oven. [From step 4a]

Req F13: The system shall stop running if the oven door is opened while the system is running. [From step 5a]

Req F14: The system shall provide a mechanism to cancel a cook time entered by Cook. [From steps 5a and 5b]

Req F15: The system shall stop running if the cook time is canceled while the system is running. [From steps 5a and 5b]

Conduct Use Case (6a)

Think about performance and other constraints to capture nonfunctional requirements and decide with stakeholders.

Example (Continued):

Req N1: The system shall allow Cook to enter the cooking time in less than five keystrokes. [Constraint on Req F1, from stakeholder interviews]

Req N2: The cooking time displayed by the system shall be visible to a Cook with 20/20 vision standing five feet from the oven in a room with an luminance level between 0 and 100 foot-candles. [Constraint on requirement F2, and from stakeholder interviews]

Req N3: The system shall raise the temperature of food in the oven such that temperatures at two distinct locations in the food are different by less than 10%. [Constraint on Req F3, and from stakeholder interviews where stakeholders desire even cooking of food]

Conduct Use Case (6b)

Think about performance and other constraints to capture nonfunctional requirements and decide with stakeholders.

Example (Continued):

Req N4: The system shall update the remaining cook time display every second. [Constraint on Req F4]

Req N5: The audible signal emitted by the oven shall have an intensity level of 80 decibels, +/- 2 decibels. [Constraint on Req F7, from stakeholder interviews]

Req N6: The system shall detect food items weighing at least 0.05 ounces and with a volume of at least 1 cubic inch. [Constraint on Req F11, obtained from stakeholder interviews]

Conduct Use Case (6c)

Think about performance and other constraints to capture nonfunctional requirements and decide with stakeholders.

Example (Continued):

Req N7: The system shall comply with section 1030 of Title 21- Food and Drugs, Chapter I – Food and Drug Administration, Department of Health and Human Services, Subchapter J: Radiological Health [Constraint on Req F3, specifies required compliance with health standards]

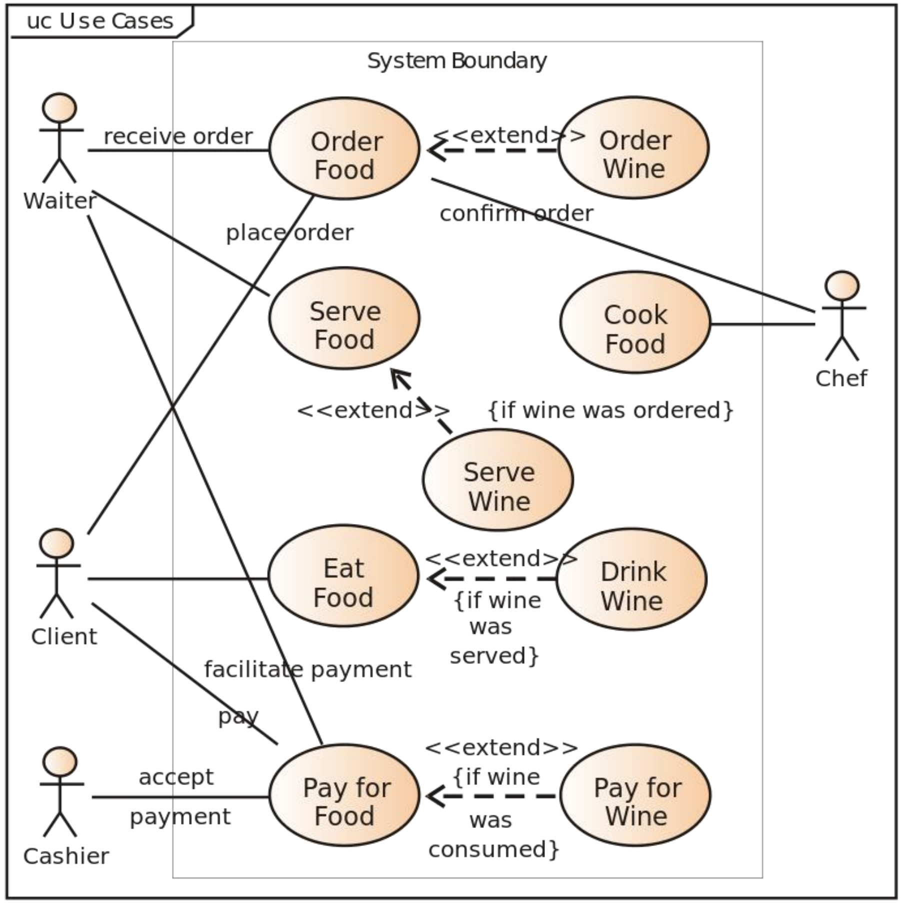
Req N8: The system shall provide 14 3/4" Width x 8 3/4" Height x 15 3/4" Depth inside cooking area volume. [From step 1, obtained by stakeholder interviews]

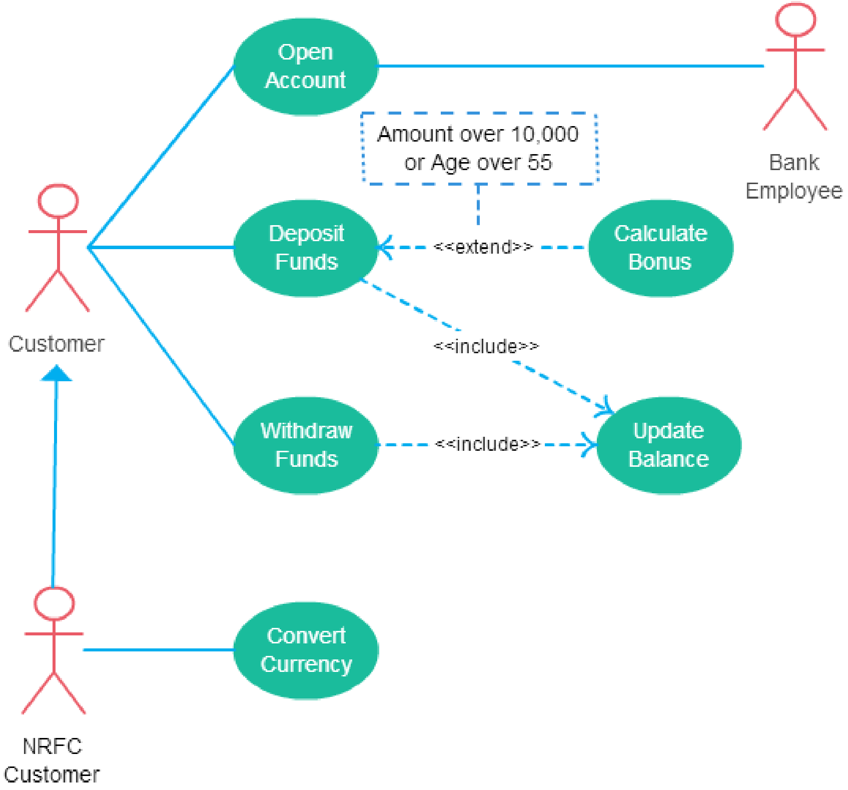
Req N9: The cooking time shall be adjustable between one second and ninety minutes [Constraint on Req F1, obtained from stakeholder interviews]

Conduct Use Case (7)

Record the captured requirements with the captured source (Use case Name and stakeholder interview if exist) to the requirements database

Use Case Examples





Requirements Engineering Process

Objectives of Requirements Engineering (RE) Process

- Requirements cannot be established without checking their impact (achievability) on lower level elements.

- Requirements definition is an iteration and balancing process that works both top-down and bottom-up.

- Once the top level requirements has been established, it is necessary to allocate and flow down to successively lower levels.

- As the allocation and flowdown is repeated, traceability to top level requirements are assured.

RE Process Variability

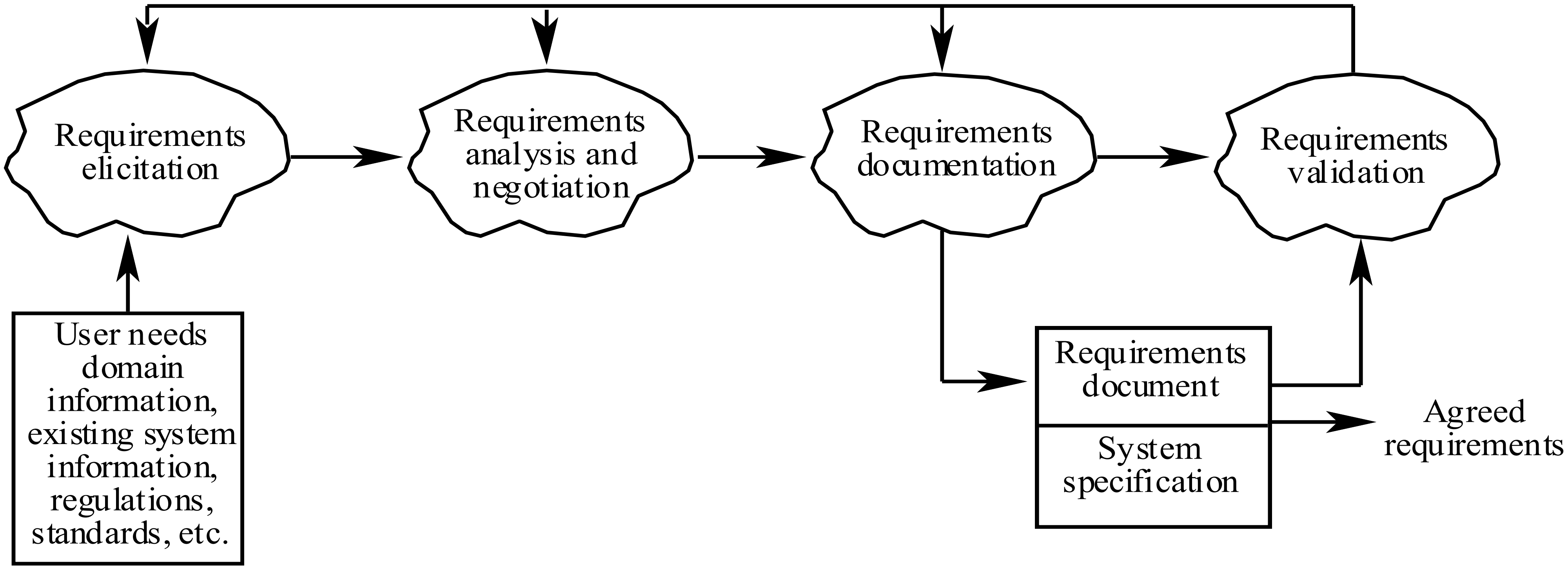
RE processes vary radically from one organization to another

Factors contributing to this variability include

- Technical maturity
- Disciplinary involvement
- Organizational culture
- Application domain

There is therefore no ‘ideal’ requirements engineering process

Example RE Process Model


Requirements elicitation
Requirements analysis and negotiation
Requirements documentation
Requirements validation
User needs domain information, existing system information, regulations, standards, etc.
Requirements document
System specification
Agreed requirements

Example RE Process Model

Requirements elicitation

- Requirements discovered through consultation with stakeholders

Requirements analysis and negotiation

- Requirements are analysed and conflicts resolved through negotiation

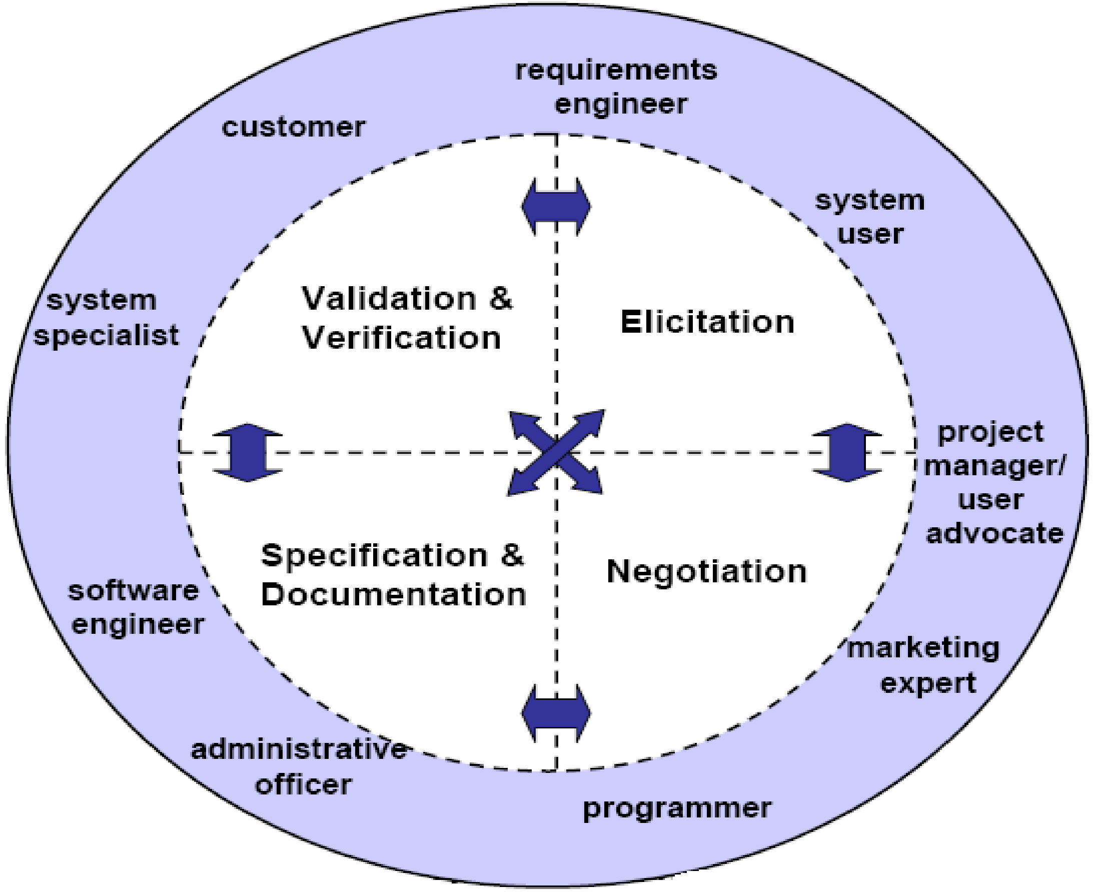
Requirements documentation

- A requirements document is produced

Requirements validation

- The requirements document is checked for consistency and completeness

Example RE Process Model



CASE Tool Support

Management tools help manage a database of requirements and support the management of changes to these requirements.

Requirements storage

- Requirements should be managed in a secure, managed data store.

Change management

- The process of change management is a workflow process whose stages can be defined and information flow between these stages partially automated.

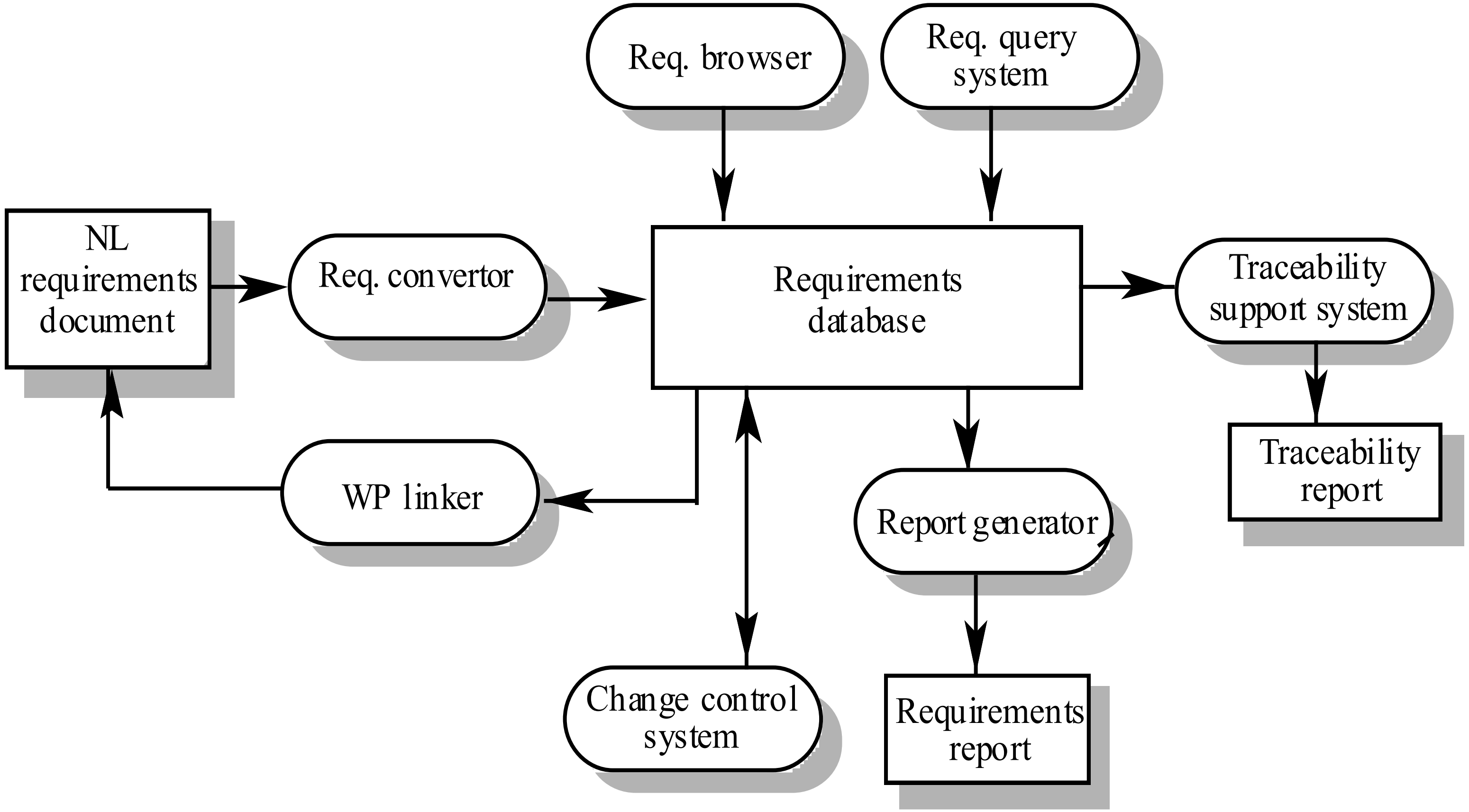
Traceability management

- Automated retrieval of the links between requirements.

Requirements Management Tools

- Requirements browser
- Requirements query system
- Requirement Traceability
- Report generator
- Requirements converter and word processor linker
- Change control system

Requirements Management Tool

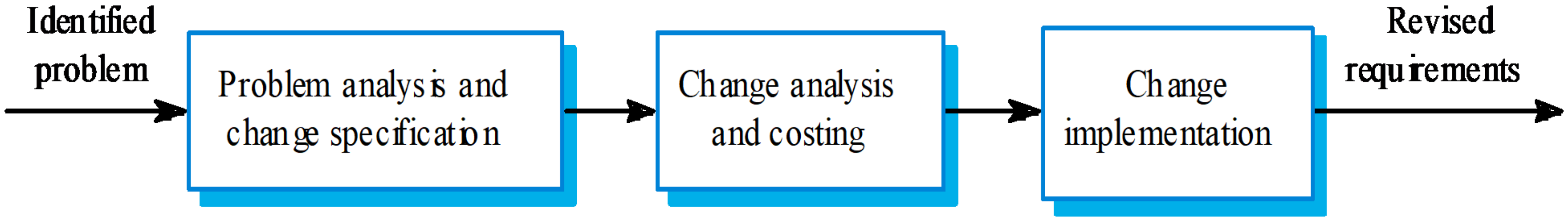


Requirements change management

Should apply to all proposed changes to the requirements.

Principal stages

- Problem analysis. Discuss requirements problem and propose change;
- Change analysis and costing. Assess effects of change on other requirements;
- Change implementation. Modify requirements document and other documents to reflect change.



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

RE Process Problems

- Lack of stakeholder involvement
- Business needs not considered
- Lack of requirements management
- Lack of defined responsibilities
- Stakeholder communication problems
- Over-long schedules and poor quality requirements documents

Requirements Reviews

- To deal with RE Process Problems regular reviews should be held while the requirements definition is being formulated
- Both client and contractor staff should be involved in reviews
- Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage

Review Check

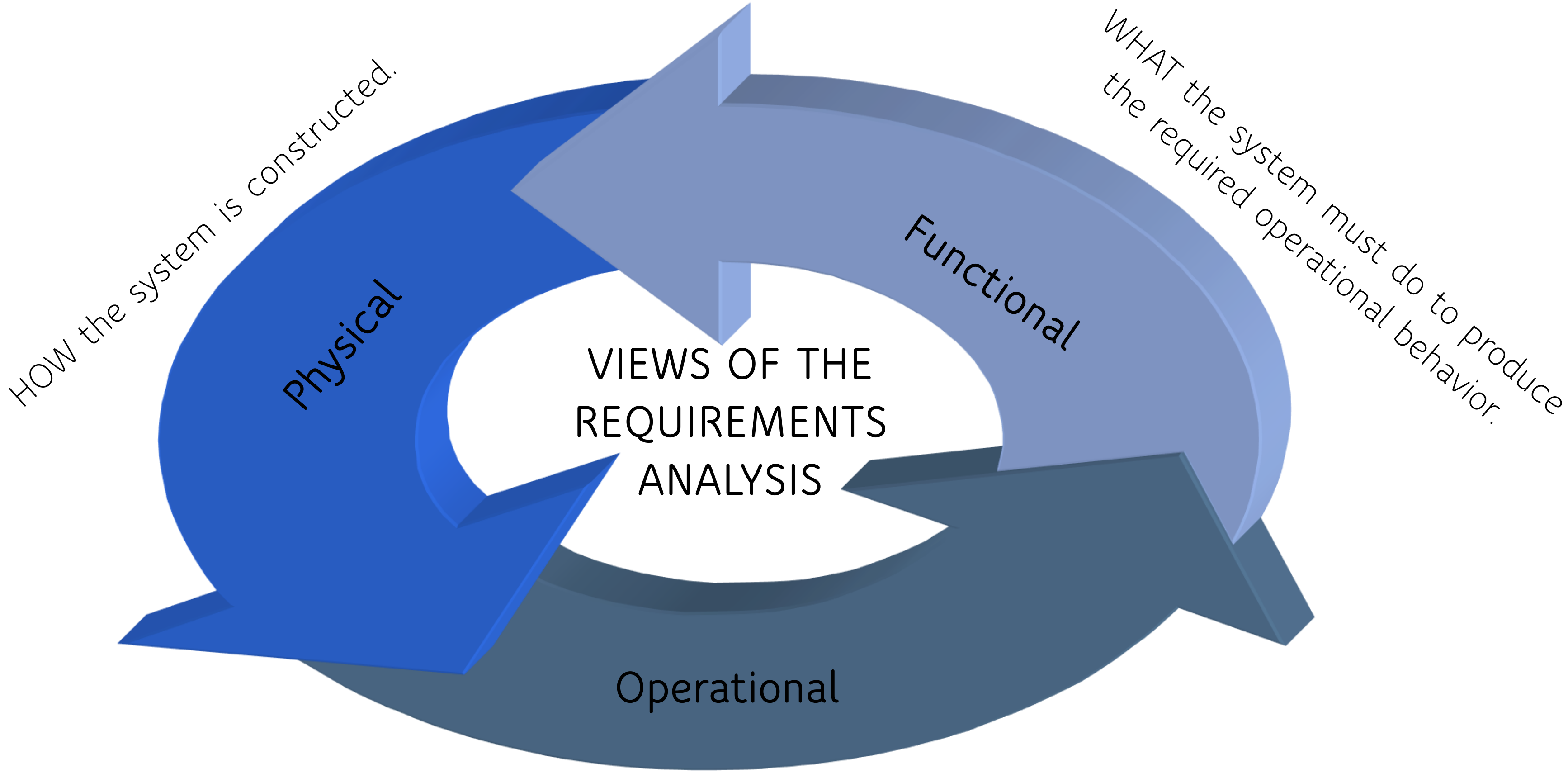
- Verifiability. Is the requirement realistically testable?
- Comprehensibility. Is the requirement properly understood?
- Traceability. Is the origin of the requirement clearly stated?
- Adaptability. Can the requirement be changed without a large impact on other requirements?

RE Process Key Points

- The requirements engineering process is a structured set of activities which lead to the production of a requirements document.
- Inputs to the requirements engineering process are information about existing systems, stakeholder needs, organizational standards, regulations and domain information.
- Requirements engineering processes vary radically from one organization to another. Most processes include requirements elicitation, requirements analysis and negotiation and requirements validation.
- Human, social and organizational factors are important influences on requirements engineering processes.
- Requirements engineering process improvement is difficult and is best tackled in an incremental way.
- Requirements engineering processes can be classified according to their degree of maturity.

Activities of The Requirements Engineering

Perspectives of View for the Requirements Analysis



HOW the system is constructed.

Physical

VIEWS OF THE REQUIREMENTS ANALYSIS

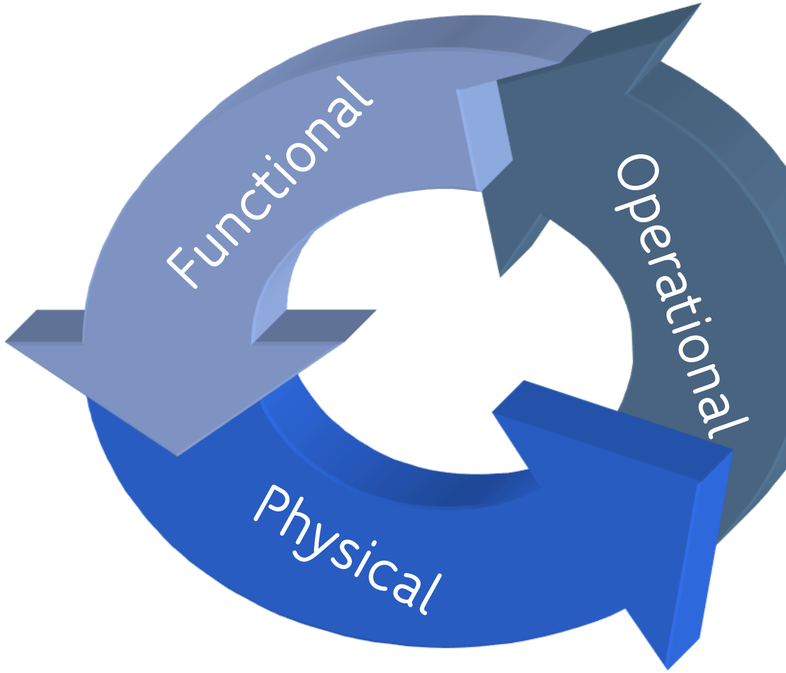
Functional

WHAT the system must do to produce the required operational behavior.

Operational

How the system will serve its users.

Operational View for the Requirements Analysis



➢Operational need definition,

➢System mission analysis,

➢Operational sequences,

➢Operational environments,

➢Conditions/events to which a system must respond,

➢Operational constraints on system,

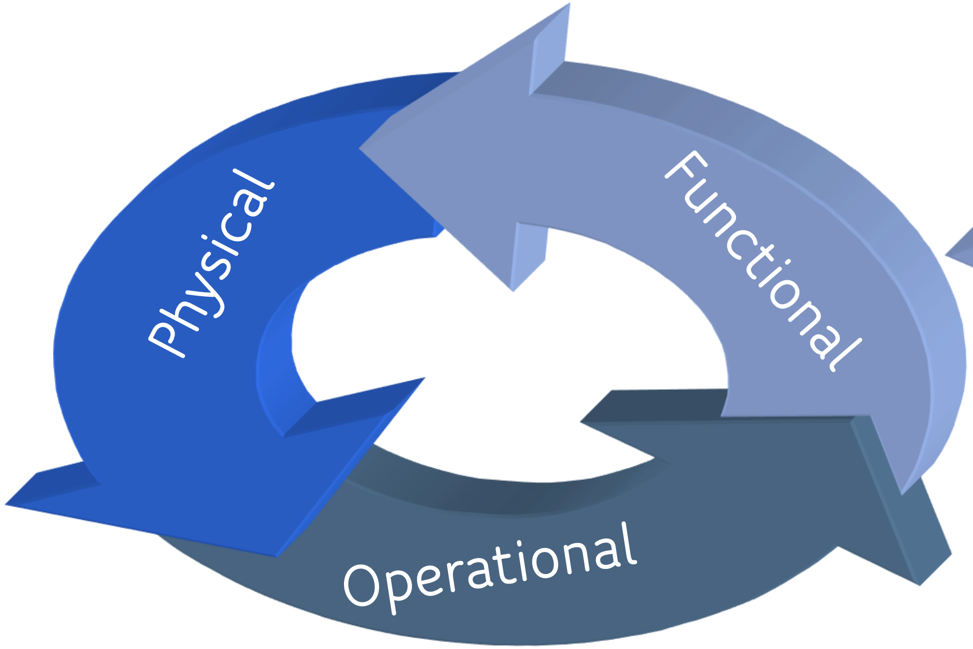
➢Mission performance requirements,

➢User and maintainer roles (defined by job tasks and skill requirements or constraints),

➢Structure of the organizations that will operate, support and maintain the system,

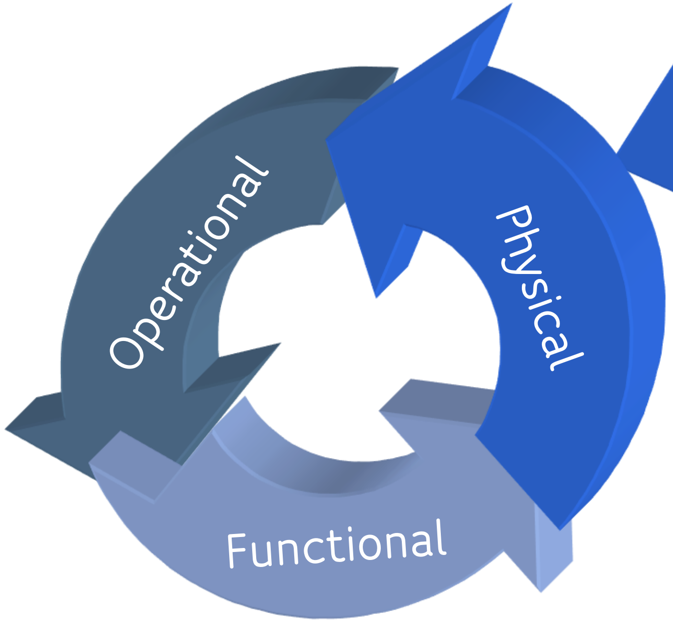
➢Operational interfaces with other systems.

Functional View for the Requirements Analysis



- ➢System functions,
- ➢System performance,
  - ✓Qualitative – how well
  - ✓Quantitative – how much, capacity
  - ✓Timeliness – how often
- ➢Tasks or actions to be performed,
- ➢Inter-function relationships,
- ➢Hardware and software functional relationships,
- ➢Performance constraints,
- ➢Interface requirements including identification of potential open-system opportunities
- ➢Unique hardware or software
- ➢Verification requirements

Physical View for the Requirements Analysis



- ➢Configuration of System:
  - ✓Interface descriptions,
  - ✓Characteristics of displays and operator controls,
  - ✓Relationships of operators to system/ physical equipment,
  - ✓Operator skills and levels to perform functions.
- ➢Characterization of Users:
  - ✓Handicaps (special operating environments),
  - ✓Constraints (movement or visual limitations).
- ➢System Physical Limitations:
  - ✓Physical limitations (capacity, size, weight),
  - ✓Technology limitations (range, precision, data rates, frequency, language),
  - ✓Government Furinished Equipment (GFE),
  - ✓Commercial-Off-the-Shelf (COTS),
  - ✓Nondevelopmental Item, reusability reqs
  - ✓Necessary or directed standards.

Concept of Operation Doc. template

I. Scope
   a) Document Overview
   b) System Overview

II. Current System or Situation
   a) Background, objectives and scope
   b) Operational policies and constraints
   c) Modes of operations

III. Justification for and nature of changes
   a) Justification of changes
   b) Description of desired changes
   c) Priorities among changes
   d) Changes considered but not included

IV. Concepts for proposed systems
   a) Background, objectives and scope
   b) Description of proposed system
   c) Operational policies and constraints
   d) Modes of operations

V. Operational Scenarios

VI. Summary of impacts
   a) Operational impacts
   b) Organizational impacts
   c) Impacts during development

VII. Analysis of the proposed system
   a) Summary of improvements
   b) Disadvantages and limitations
   c) Alternatives and trade-offs considered

Project Management Plan doc. Template

1 INTRODUCTION
1.1 Purpose of Project Management Plan
2 EXECUTIVE SUMMARY OF PROJECT
2.1 Assumptions/Constraints
3 SCOPE MANAGEMENT
3.1 Work Breakdown Structure
3.2 Deployment Plan
3.3 Change Control Management
3.4 Project Deliverables
4 SCHEDULE/TIME MANAGEMENT
4.1 Milestones
4.2 Project Schedule
4.2.1 Dependencies

5 COST/BUDGET MANAGEMENT
6 QUALITY MANAGEMENT
7 HUMAN RESOURCE MANAGEMENT
7.1 Project Organizations
7.2 Roles & Responsibilities
8 COMMUNICATIONS MANAGEMENT
8.1 Communication Matrix
9 RISK MANAGEMENT
9.1 Risk Management
9.2 Risk Database
10 ISSUE MANAGEMENT
10.1Issue Management
10.2 Issue database
11 PROCUREMENT MANAGEMENT

SRS Document Template

I. INTRODUCTION

- a) Document Purpose
- b) Product Scope & Overview
- c) Intended Audience & Document Overview
- d) Document Conventions
- e) References & Acknowledgments

II. OVERALL DESCRIPTION

- a) Product Overview
- b) Product Functionality
- c) Design and Implementation Constraints
- d) Assumptions and Dependencies

III. SPECIFIC REQUIREMENTS

- a) External Interface Requirements
- b) Contex Model
- c) Functional Requirements
- d) Use Case Model

IV. OTHER NON-FUNCTIONAL REQUIREMENTS

- a) Performance Requirements
- b) Safety, Reliability Requirements
- c) Security Requirements
- d) Software Quality Attributes

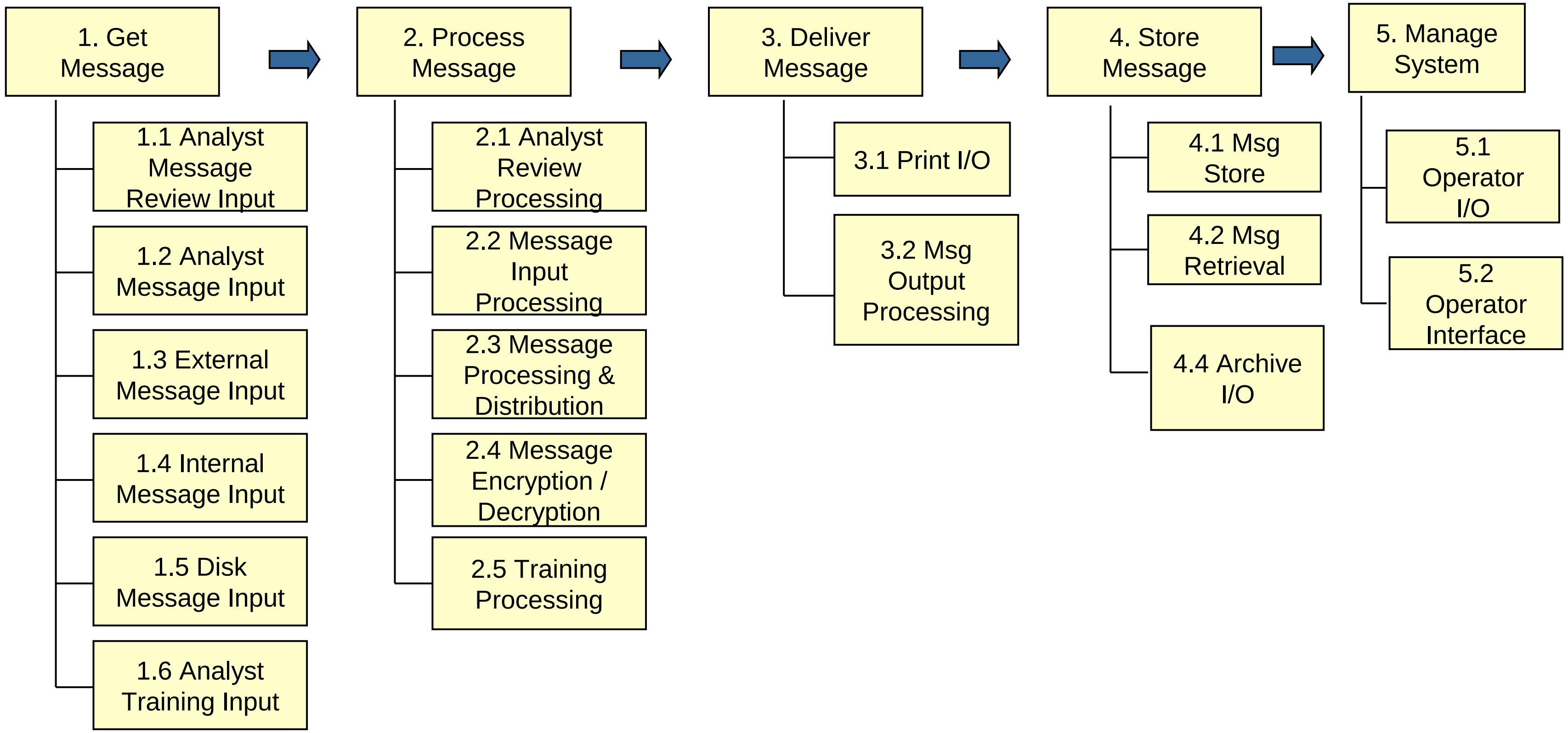
V. OTHER REQUIREMENTS

- a) Usage Requirement
- b) Efficiency Requirements
- c) Resources Requirements

VI. REQUIREMENTS RATIONALE

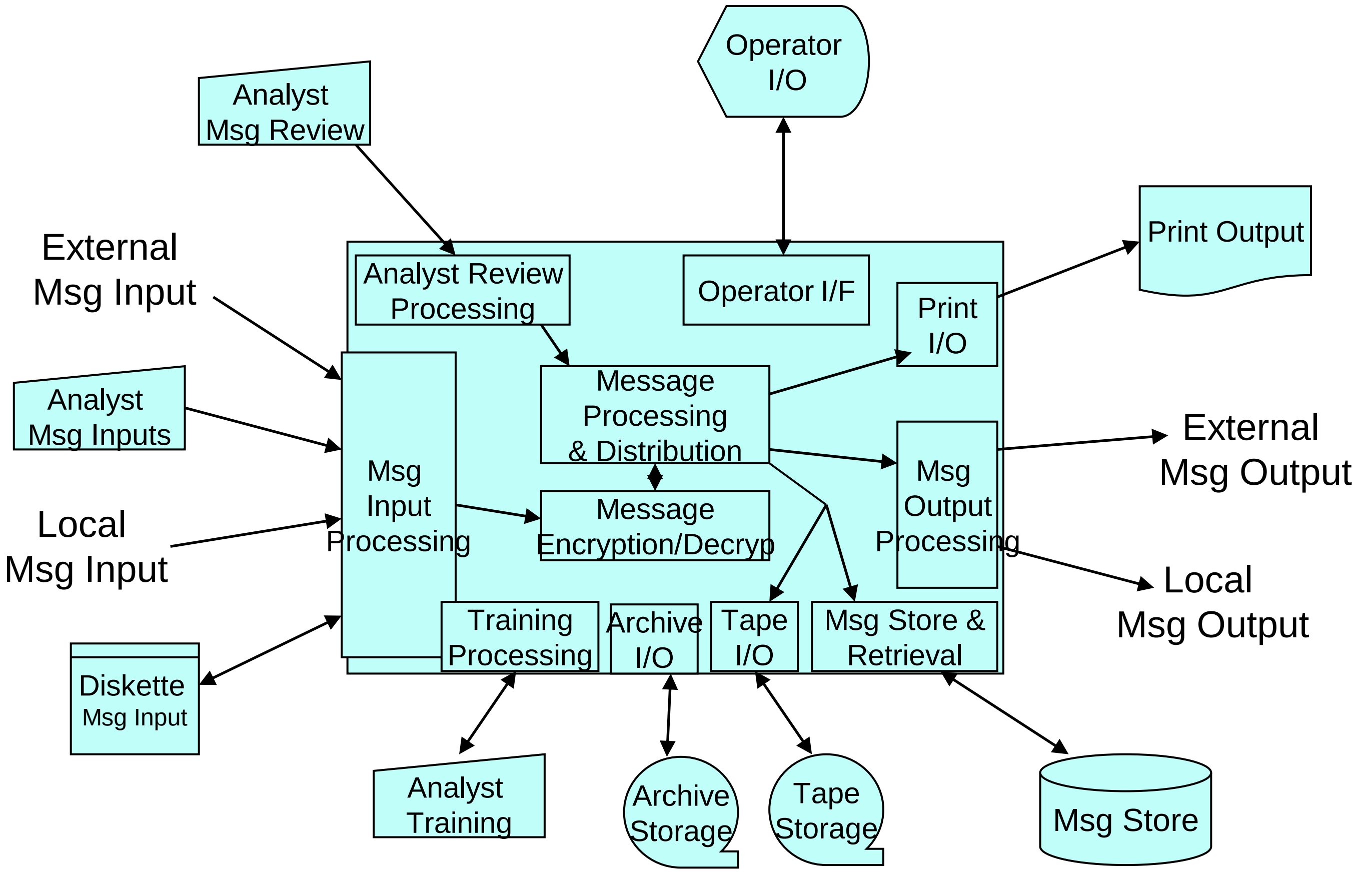
Quiz Solution

Functional View of MPS



- 1. Get Message
  - 1.1 Analyst Message Review Input
  - 1.2 Analyst Message Input
  - 1.3 External Message Input
  - 1.4 Internal Message Input
  - 1.5 Disk Message Input
  - 1.6 Analyst Training Input
- 2. Process Message
  - 2.1 Analyst Review Processing
  - 2.2 Message Input Processing
  - 2.3 Message Processing & Distribution
  - 2.4 Message Encryption / Decryption
  - 2.5 Training Processing
- 3. Deliver Message
  - 3.1 Print I/O
  - 3.2 Msg Output Processing
- 4. Store Message
  - 4.1 Msg Store
  - 4.2 Msg Retrieval
  - 4.4 Archive I/O
- 5. Manage System
  - 5.1 Operator I/O
  - 5.2 Operator Interface

MPS Functional View



MPS System Req. Spec.

Req 1- MPS Systems shall provide to MPS Analysts to review GMCS messages to determine if the correct classification was made.

Req 2- MPS system shall operate 24 hours per day in an automated mode to process either incoming or outgoing messages on the GMCS.

Req 3- MPS System shall provide storage capacity that last 60 days of all local messages are to be stored (less than 10 TB).

Req 4- MPS system shall provide tape storage for messages over 60 days old for achieving.

Req 5- MPS shall provide a capability to encrypt/decrypt messages with secret classification.

MPS System Req. Spec.

Req 6- MPS System shall provide an analyst training capability which can be used during normal operations without disturbing message processing.

Req 7- MPS System shall provide an analyst training capability as given below:

a) Create a message
b) Retrieve a message
c) Edit a message
d) Save/delete a message
e) Security classification for a message….

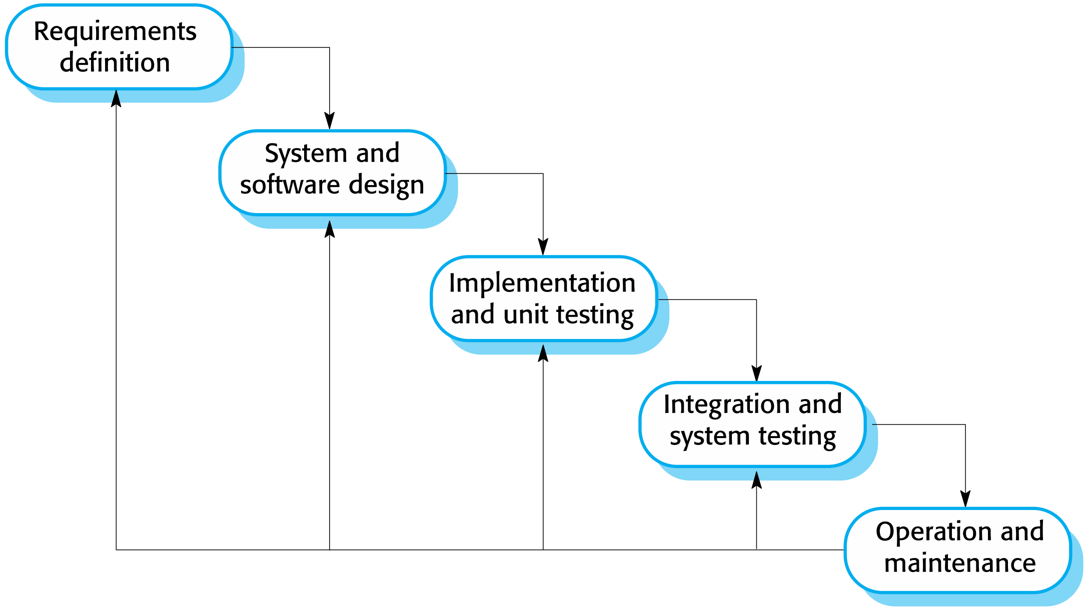
SW Processes

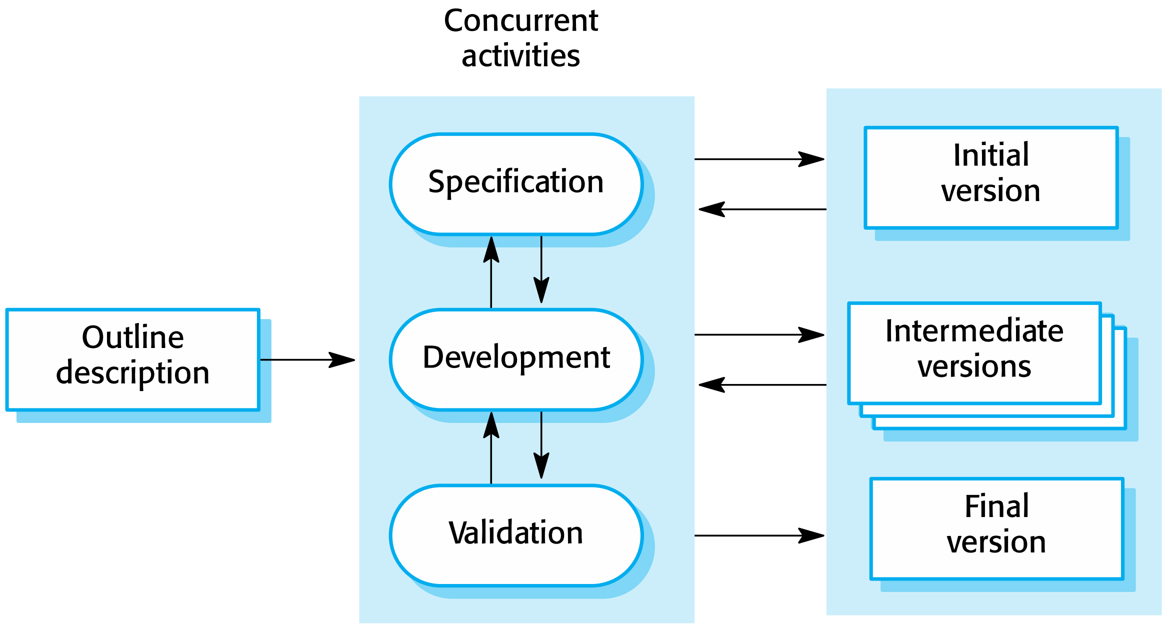
SW Processes; structured set of activities required to develop a software system.

Many different software processes but all involve:

- Specification – defining what the system should do;
- Design and implementation – defining the organization of the system and implementing the system;
- Validation & Verification – checking that it does what the customer wants;
- Evolution – changing the system in response to changing customer needs.

The waterfall model vs incremental model


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance


Concurrent activities
Outline description
Specification
Development
Validation
Initial version
Intermediate versions
Final version