Requirements Elicitation and Management, Use Cases

Lecture 3
3 April 2025

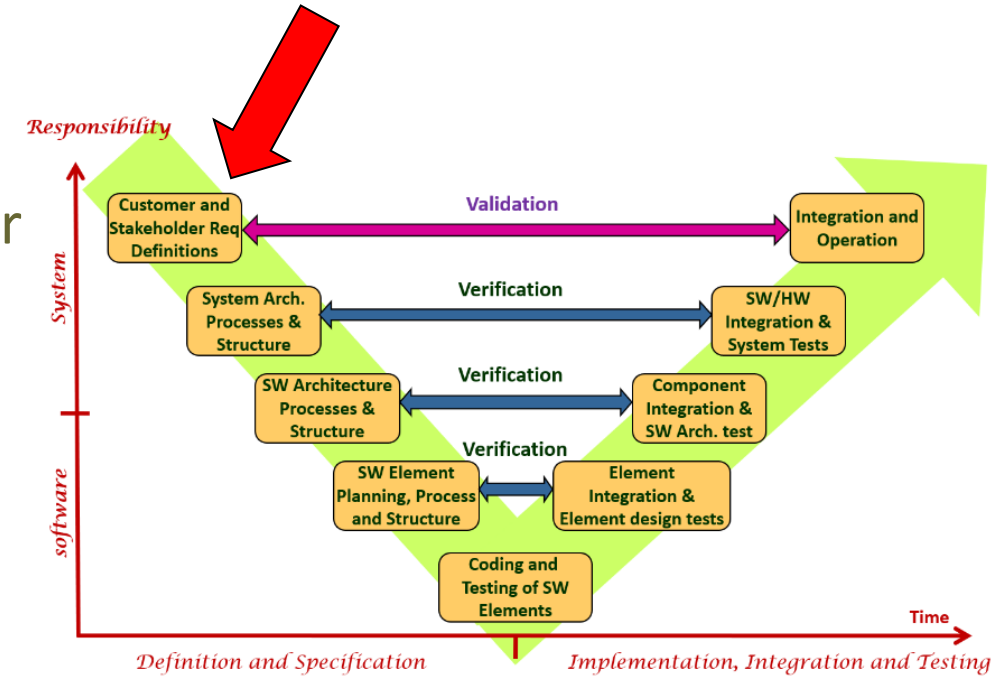
Slides created by
Prof Amir Tomer
tomera@cs.technion.ac.il

Topics for Today

- Requirements Elicitation and Management
- Use Cases
  - Identifying system use cases

Eliciting requirements from customers and stakeholders

- Goal of the stage
  - Understand the problem/need + the solution concept
  - Collect, concretize, categorize requirements
    - Reflect the parts of the problem/need and the solution concept
    - Are the basis for managing the development
- Input
  - Technical design, operational design (i.e. the user stories and stakeholder stories)
- Result
  - Categorized requirements table



Event Analysis – Online Voting Systems

1. 2 December 2008: Primaries for Israel Labor (עבודה) Party
   - Fault: Router communications failed under load
   - Result: Cancelation of online voting and repeat with manual voting
2. 9 December 2008: Primaries for Israel Likud Party
   - Fault: Voters had difficulty completing the voting process, leading to long lines
   - Result: Voting places remained open until 01:00AM
3. 25 November 2012: Primaries for Israel Likud Party
   - Fault: Miscellaneous errors (server failures, stuck computers)
   - Result: Voting extended to the following day

Did the voting system meet its requirements?

- Below are some formal requirements for online voting systems.
  - Assembled by the high-level IT committee of the Ministry of Finance (Or Hirshona, The Marker, 11 Dec 2008)

Filtering/Dividing Requirements

Sentences and phrases found in the specification may contain or imply different types of requirements

> In “automatic scan” mode, the multifunction device automatically identifies the source document type – photograph, business card, letter – and automatically scans it, storing the scan in memory. This all occurs without user intervention

Requirements included or implied:

Operational	Data	QA: Usability
Device identifies	Automatic scan mode	Without user intervention
Device scans	Photograph document type	
Device stores	Business card document type	
	Letter document type	

Exercise: Categorizing and Dividing Requirements

- For each of the following sentences, identify the requirements in the sentence and write down how many of them there are in each category

#	Sentence	FR		NFR				
		OR	DR	PR	QA	HC	IC	MC
1	The system includes a central server that offers 24/7 service to users who access it via the internet using a commercial browser							
2	The app will operate correctly under iOS, Android, and Windows Phone							
3	The system will be able to recover from crashes within 2 minutes from restart. All data will be restored from the last hourly backup.							
4	The self driving car will sample the camera every 50ms and apply the brakes no more than 1s after it identifies an obstacle on the road							
5	The system prototype, based on the Oracle DBMS, will be provided within 3 months of the signing of the contact.							
6	After successful authentication, the ATM shows the main screen that includes options for withdrawal, transfer between accounts, and display balance. A red “cancel” button will be shown in the bottom left of the screen.							
7	The navigation device creates the route, gives directions, and remembers the selected destinations. The car’s automatic navigation capability helps users avoid getting lost and gets them to their destinations easily.							

What do we require of Requirements?

Discrete and Identified

- Readable as a complete and independent sentence
  - Too complex → break into pieces
- 1-to-1 numbering of each requirement
  - ID remains throughout the development, even if the requirement is canceled

Unambiguous

- Ensure people agree what it means

Complete

- Requirements must cover all aspects well (beware of TBD)

Consistent

- Ensure no contradictions

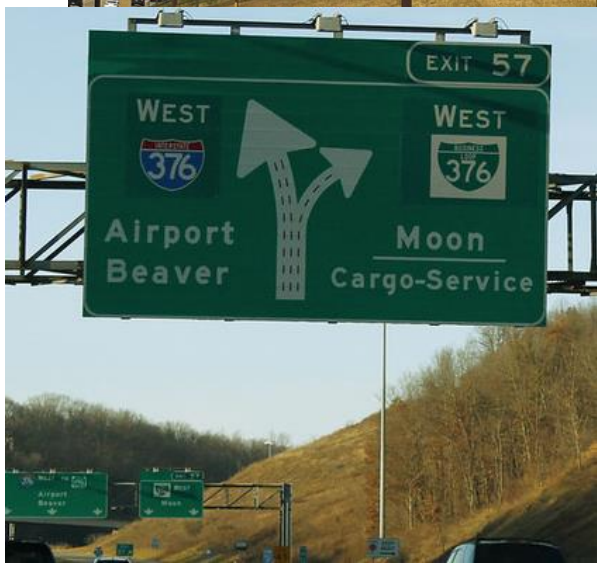
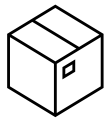


Image sources: https://www.flickr.com/photos/formulanone/27419842630/, https://www.aaroads.com/forum/index.php?topic=19450.275

What do we require of Requirements?

Traceable to origin

- Where did it come from? Explicit? Derived?

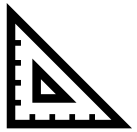


Not design the system on their own (what, not how)

- Choice of specific algorithm, component, etc. reduces design options
- Ensure any such one is truly necessary

Testable and measurable

- Specify how the requirement will be measured to ensure it's met
  - Analysis, simulation
  - Lab testing
  - Operational testing



Requirements table: Basis of managing development

ID	Requirement	Source	Type	Additional Attributes
	…			

- ID: Unique identifier, kept even if the requirement is canceled
- Requirement: Clear, unambiguous, accurate to the source
- Source: Which stakeholder or document introduced it
- Type: FR, NFR and sub-category
- Additional attributes: (next)

Additional Attributes

Priority/Importance

- Critical/Non-Critical

Difficulty/Technical Risk

- Developer experience needed to implement

Cost

- Not necessarily correlates with difficulty

Owner/Responsibility

- Assign to team, developer, partner, subcontractor

Rationale (Why? Source of values)

- Mostly to avoid investing too much in overspecification

Testing/Proof Method

- Similarity, analysis, simulation, lab test, field test, pilot, operational test

Status

- Proposed, accepted, analyzed, assigned, implemented, tested

(and so on)…

Partial requirements for elevator system

ID	Requirement	FR	NFR
1	The system includes 3 elevators	DR	HC
2	The system serves 10 floors	DR	
3	Every floor, except the bottom and top floors have two buttons – up and down	DR	HC
4	On the bottom floor there is only an up button	DR	HC
5	On the top floor there is only a down button	DR	HC
6	A passenger on any floor … can press on the button for the desired direction	OR	
7	The elevator has an emergency stop button		HC
8	The elevator has a rescue call button		HC
9	The system will be handicapped accessible		QA

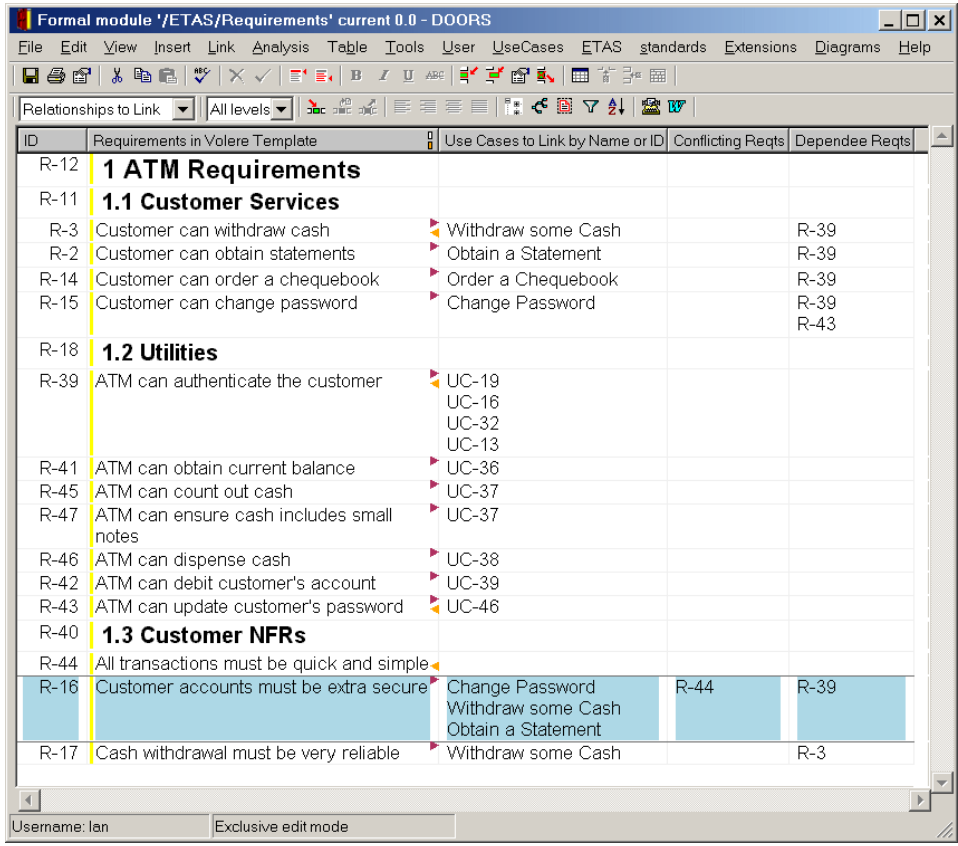
Assignment: Build a requirements table

- Use the ePark Client story and fill in the requirements table template based on it
  - Stick to the original language
    - Break the story into lines and then move the requirements to the table
  - Edit the language to make each requirement a complete and independent sentence
- Put your results in the Moodle forum for today

3 April 2025

Software tools for requirements management

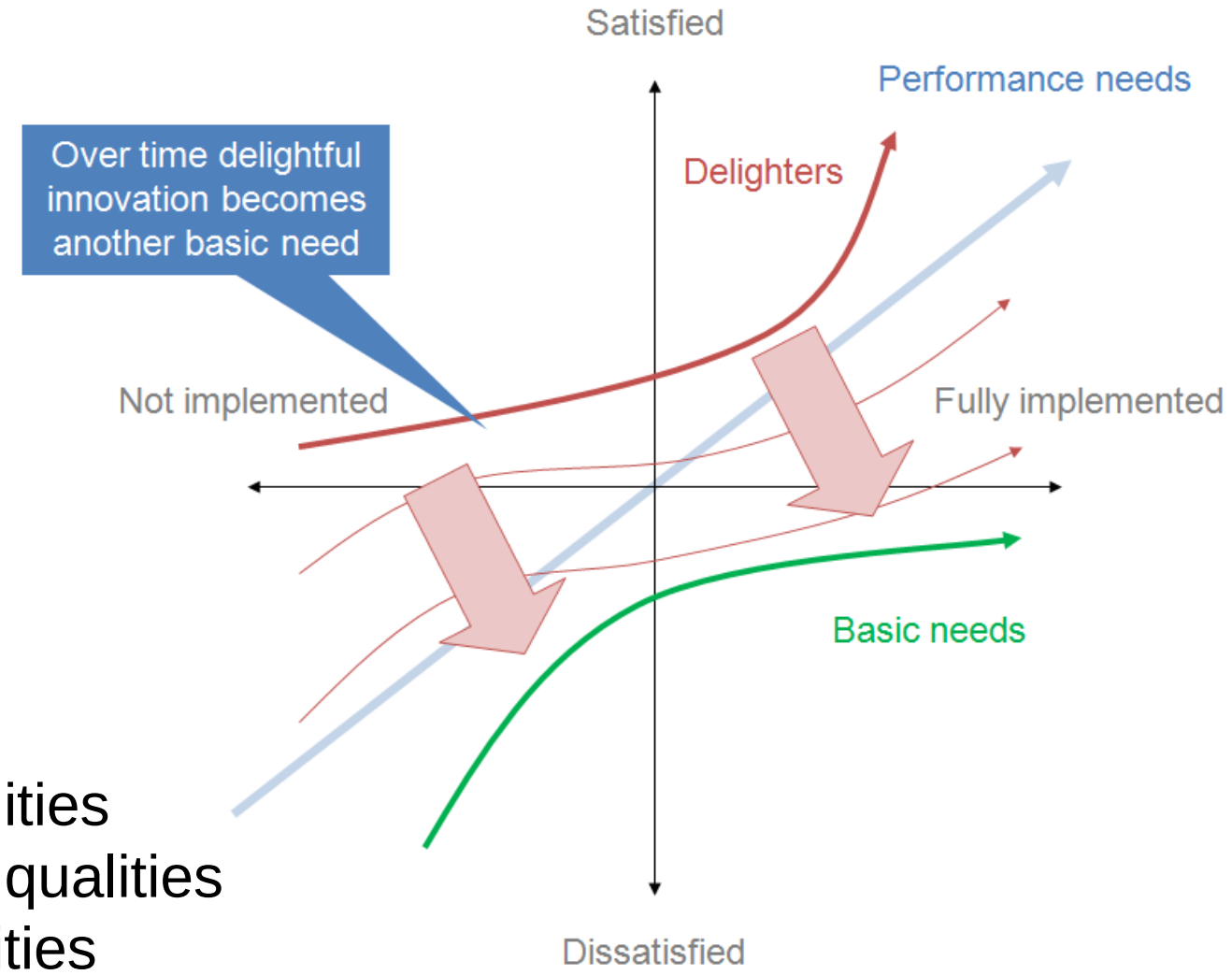
- Requirements management throughout the development lifecycle can be managed with tables and attributes
- There are some software aided requirements management tools (e.g. Doors from IBM)
  - Hierarchical requirements tables
  - Connections between requirements
  - Traceability between requirements and source documents
  - Track requirements to specification elements
  - Validate and Verify tables for testing and experiments
  - Create textual requirements lists
  - Managing requirements versions over time



ID	Requirements in Volere Template	Use Cases to Link by Name or ID	Conflicting Reqts	Dependee Reqts
R-12	**1 ATM Requirements**			
R-11	**1.1 Customer Services**			
R-3	Customer can withdraw cash	Withdraw some Cash		R-39
R-2	Customer can obtain statements	Obtain a Statement		R-39
R-14	Customer can order a chequebook	Order a Chequebook		R-39
R-15	Customer can change password	Change Password		R-39 R-43
R-18	**1.2 Utilities**			
R-39	ATM can authenticate the customer	UC-19 UC-16 UC-32 UC-13		
R-41	ATM can obtain current balance	UC-36		
R-45	ATM can count out cash	UC-37		
R-47	ATM can ensure cash includes small notes	UC-37		
R-46	ATM can dispense cash	UC-38		
R-42	ATM can debit customer's account	UC-39		
R-43	ATM can update customer's password	UC-46		
R-40	**1.3 Customer NFRs**			
R-44	All transactions must be quick and simple			
R-16	Customer accounts must be extra secure	Change Password Withdraw some Cash Obtain a Statement	R-44	R-39
R-17	Cash withdrawal must be very reliable	Withdraw some Cash		R-3

Image source: https://www.scenarioplus.org.uk/papers/traceability/relationships_to_link_DOORS_module.gif

Customers and Requirements: Kano Model



Discern:

- Must-have qualities
- Linear/Regular qualities
- Delighting qualities

Image credit: By Craigwbrown - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=23262780

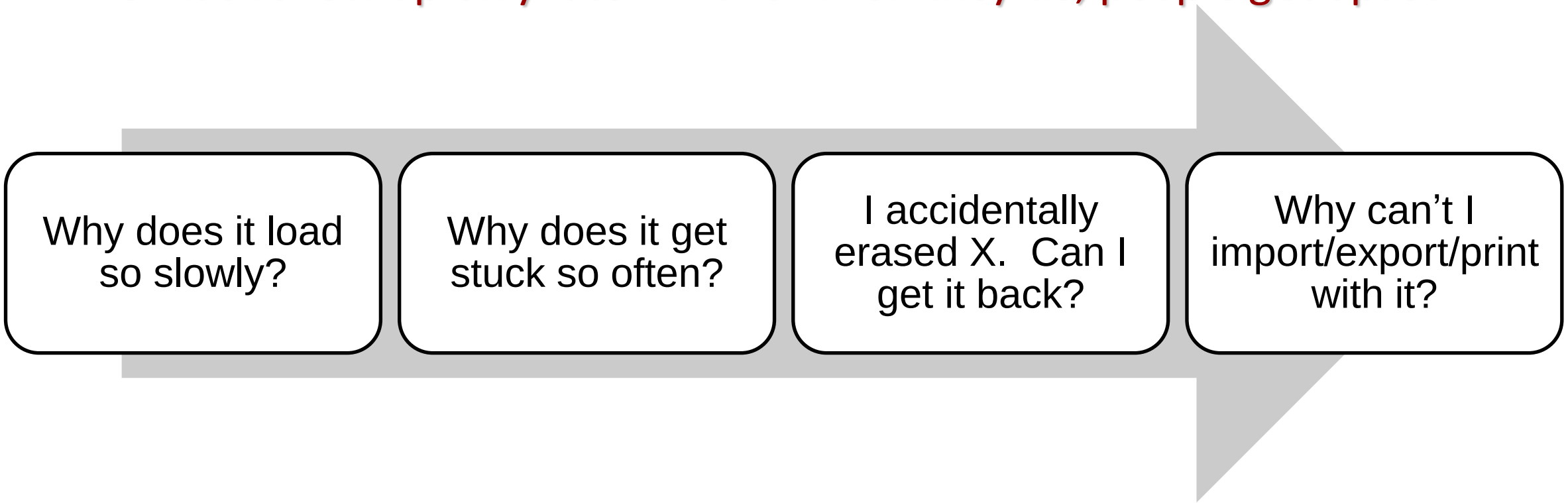
Tacit “Must Have” qualities

- Tacit requirements
  - It’s possible to run the elevator in “Sabbath” mode (a functional requirement from other stakeholders)
  - Elevator stops smoothly (safety)
- Delighting requirements
  - Each floor includes a clock that shows how long until the elevator car arrives (functional and design requirement)
  - Elevator can operate via cell phone command (usability)

Image source: https://marcellusdrilling.com/wp-content/uploads/2015/01/but-wait-theres-more.jpg

Quality Requirements: Tacit Requirements

- Most system specifications are functional
- Non-functional requirements (quality attributes) are tacit requirements
- Their lack show up only later → and when they do, people get upset



Quality Requirements: Tacit Requirements

- Work out some tacit requirements by asking “How well?” questions about functional aspects
- **Example FR:** The navigation app will compare the car’s location from the GPS with the route on the map. If a change of route is detected, it will recalculate the route.

What happens if the GPS has no signal?	How precise is the comparison?	Can the user cancel the recalculation?

- The questions will lead to addition/division of requirements into NFR and additional FR

If the GPS has no signal for X seconds, the app will warn the user	If there is no GPS signal, the app will approximate the location based on cellular tower signals.

3 April 2025

Implementing Requirements

Functional Requirements

- Assign the requirement to a specification or design element that will provide the required functionality
- Example: algorithm, database, function

Non-Functional Requirements

- Choose implementation/design options that ensure the it's fulfilled while meeting functionality

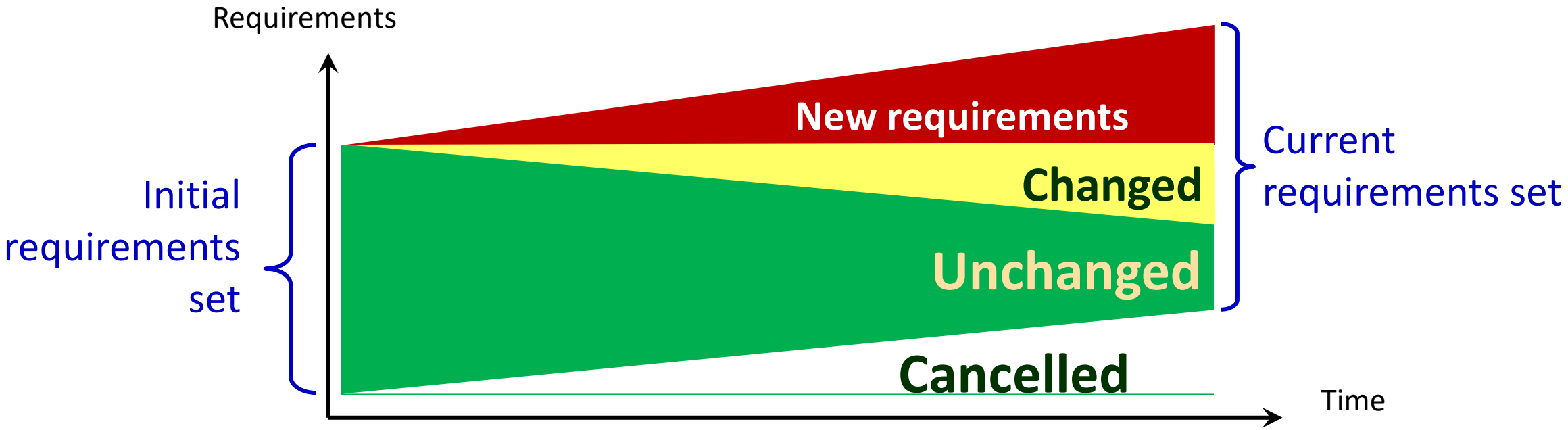
Choose an algorithm that saves space/time to meet performance requirements	Choose a multi-server architecture to meet availability requirements

- Derive FR that will lead to the NFR being met

Add a FR encryption requirement to meet security requirements	Add specific FR to meet usability requirements

Evolution of Requirements during the project

- Requirements normally are not static during development
  - New, changed, removed
- The effects of this are
  - Engineering: Design and implementation changes due to requirements changes
  - Management: Project planning (cost, time) change due changes in underlying requirements



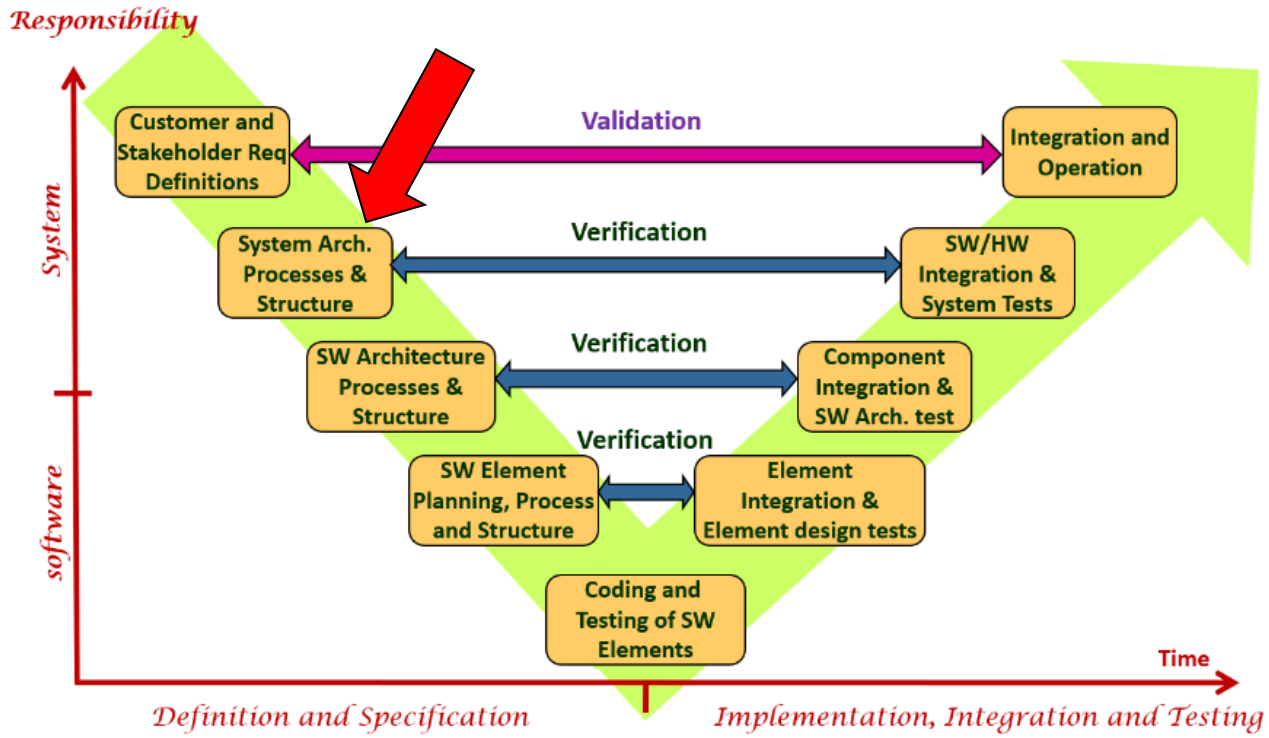
Source: Systems Engineering Handbook, INCOSE

So far

- Requirements Elicitation and Management
- Use Cases
  - Identifying system use cases

Identifying the system processes

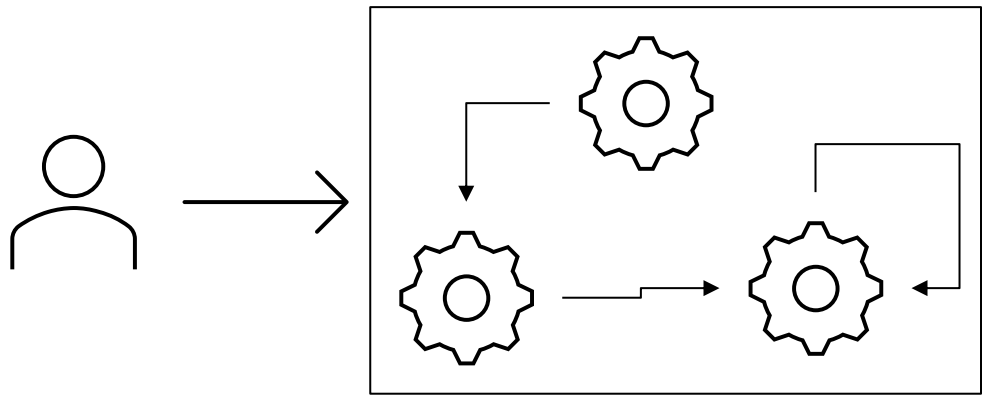
- Our goal: identify the processes in the system (system services) and the environment in which it operates
  - And how they interact!
- Input: Technical specification, operational specification, requirements table
- Output: Use Case diagram



Use Cases: Define system goals and how they are achieved

Why?

- Purpose of the system: serve the aided environment
  - Operators, users, other systems



What?

- Requirements define the properties and abilities the system needs to fulfill its purpose
- In SIS, they are expressed via use cases
  - Interaction with the external environment
  - Internal processes
  - Interaction between internal components

Use Cases: Define system goals and how they are achieved

- Operational specification (user story) defines how the system works
- Examples:
  - Start a new trip (choose destination, prepare routes, choose best routes)
  - Navigate (track vehicle location, show the map and route, recognize changes and reroute)
  - Report on obstacles (where, what time, get details)

Use cases show how the system performs the parts of the user story

Identifying Use Cases in User Story

A passenger who is on a particular floor and wants to call an elevator presses the appropriate button for the direction of travel (up or down). If the button was not already lit, the button lights up after being pressed. An elevator car traveling in the requested direction will arrive at the floor within one minute at most. When the car arrives, the door opens and the light on the button turns off.

A passenger who is in an elevator car and wants to travel to a particular floor presses on the button associated with the desired floor. If the button wasn't already lit, the button lights up and a new stop request for the floor is registered. The door closes. After a short pause, the car continues moving. It stops at every floor for which there is a stop request. When the car stops at a floor, the door opens and the button for the floor turns off.

A passenger can stop the elevator car when its moving by pressing on the emergency stop button. In that case, the elevator stops immediately and all registered stop requests are erased. Afterwards, the elevator can resume operation by pressing a button for any floor.

Identifying Use Cases in User Story

If the elevator gets stuck while in operation, passengers can call for help using the emergency rescue button. The rescuer (the building maintenance engineer) will access the emergency panel in the machine room and perform operations to move the elevator car to the bottom floor and open the door.

The maintenance engineer is responsible to start up the elevators at the beginning of the day and to shut it down at the end of the day. The technician comes once every 6 months and performs a complete check of the system and fixes problems using the technician panel in the machine room.

The elevator system must meet all applicable safety rules.

The system must be accessible to the handicapped.

System use case table

- Write down the system use cases
  - For every use case, mark who initiates it and what is the desired goal
    - We'll fill in the rest later
- Sample table for the elevator story

Initiator	Use Case	Goal
Passenger	Call elevator	Get an elevator ready for travel
Passenger	Ride elevator	Arrive at the desired floor (including emergency stop as needed)
Passenger	Rescue	Rescue from a stuck elevator
Technician	Inspection	Check the proper operation and serviceability of the elevator, find problems
Technician	Repair	Return elevator to proper state and usability
Maintenance Engineer	Start up	Start up the system
Maintenance Engineer	Shut down	Turn off the system

Assignment: Identify use cases in ePark story

- In the ePark story, mark down text that signifies use cases
- Build a table in the following format

Initiator	Use Case	Goal

Conclusion

- Requirements Elicitation and Management
- Use Cases
  - Identifying system use cases