SOFTWARE REQUIREMENT

SWR - BinhNN7

About me

Nguyễn Nguyên Bình - 1981

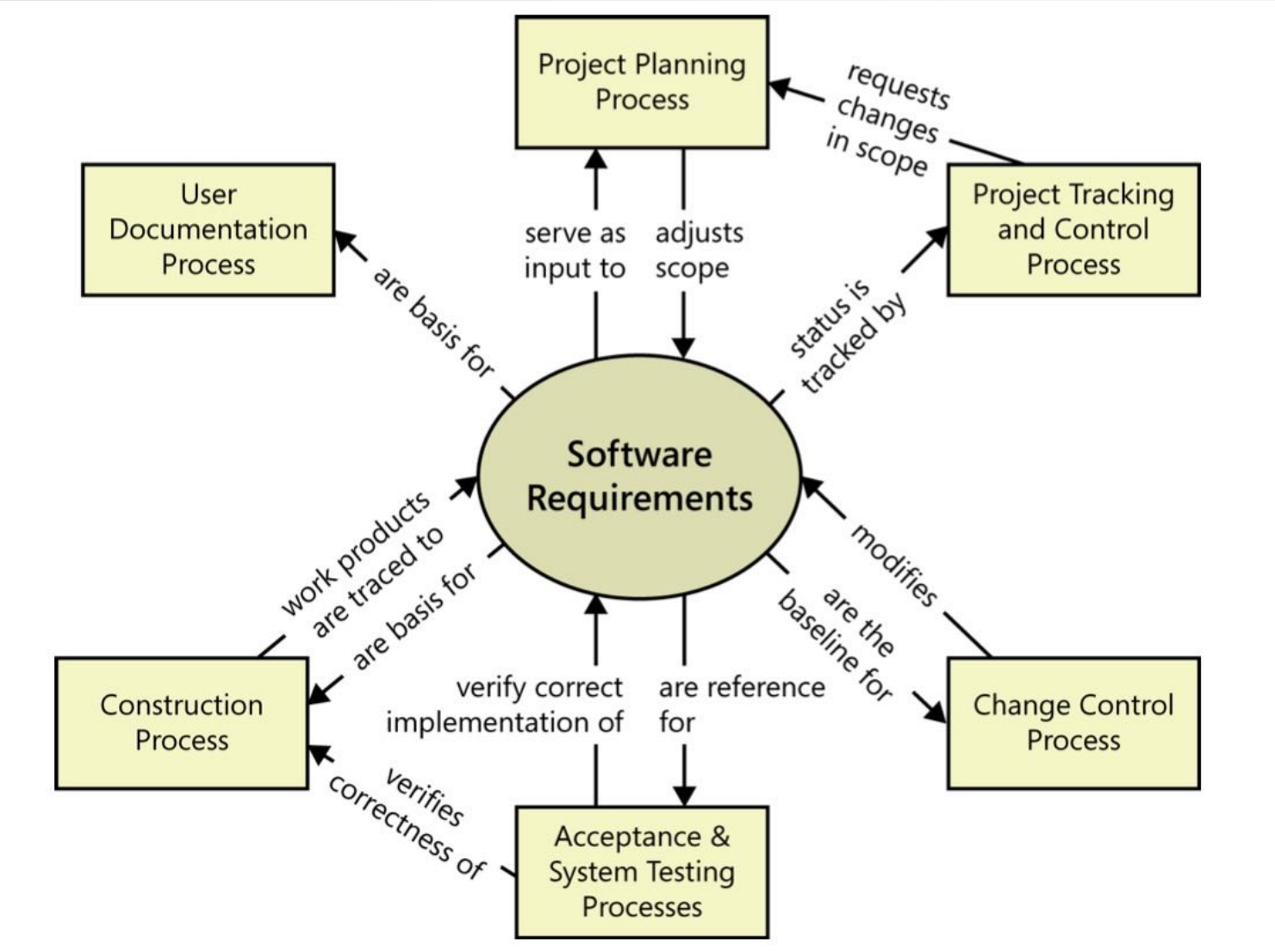
- Master of Science in Computer Science from Budapest University of Technology and Economics
- Work in IT area from 2002 – present
  - 2002- 2006: SE - Device firmware development
  - 2006-2009: SSE - Grid computing
  - 2009-2012: BA/PM - Banking services development & integration (card payment system, core-banking, internet / phone banking,…)
  - 2012-2016: Delivery Manager - Outsourcing (e-commerce, stock trading, healthcare, IoT,…)
  - 2016-present: CTO/Product Director - IoT, Data analyzing & Machine learning, Video Conference
- Education area
  - 2018 – present: guest lecturer of FPT University

Assignment

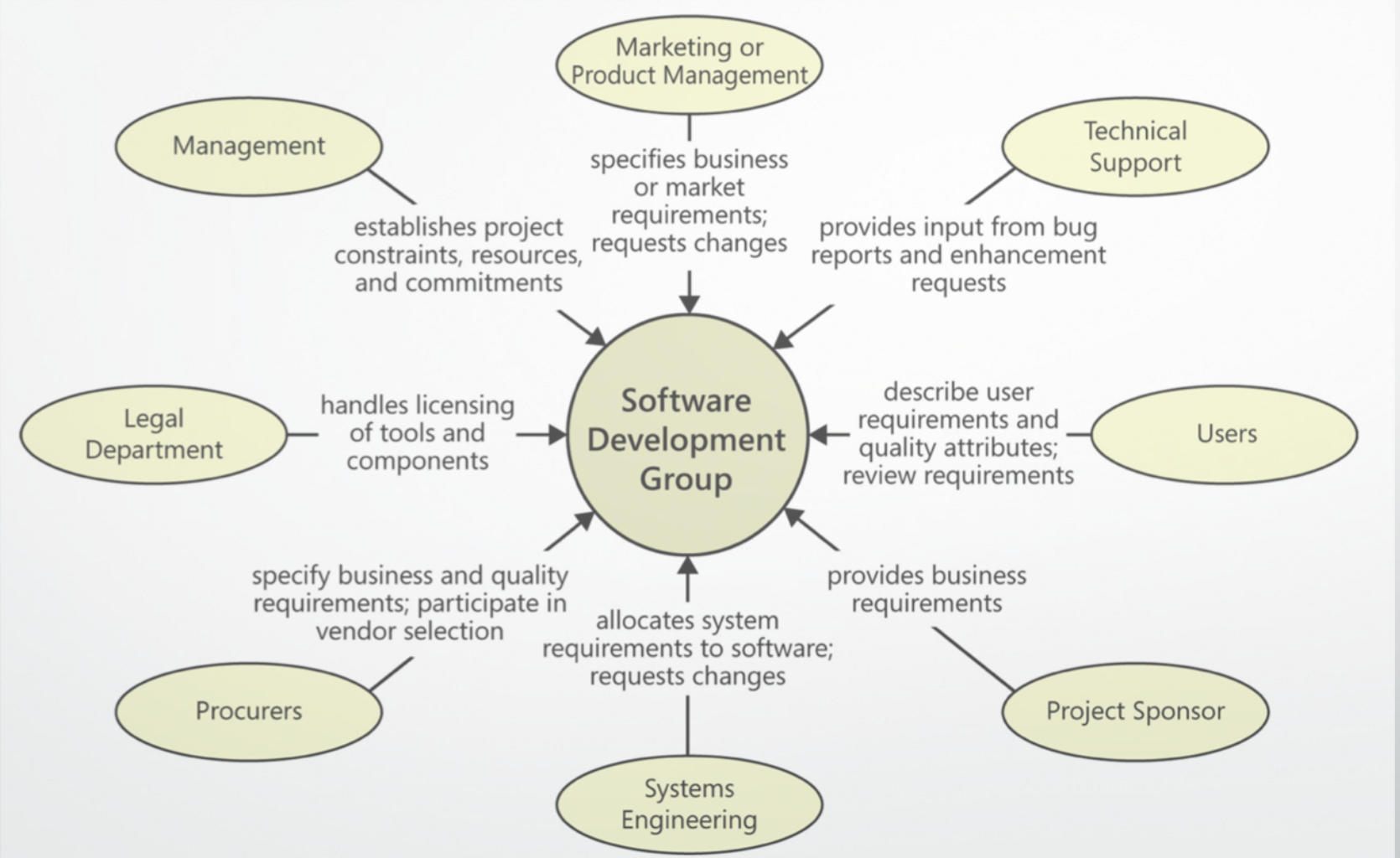
- Quizes: 3 quizzes
- Assignment:
  - Individual
  - Diagrams and req.
- LMS – SWR302 (BinhNN7) – pass: octopus6wolf
- Email: binhnn7@fe.edu.vn

SDLC

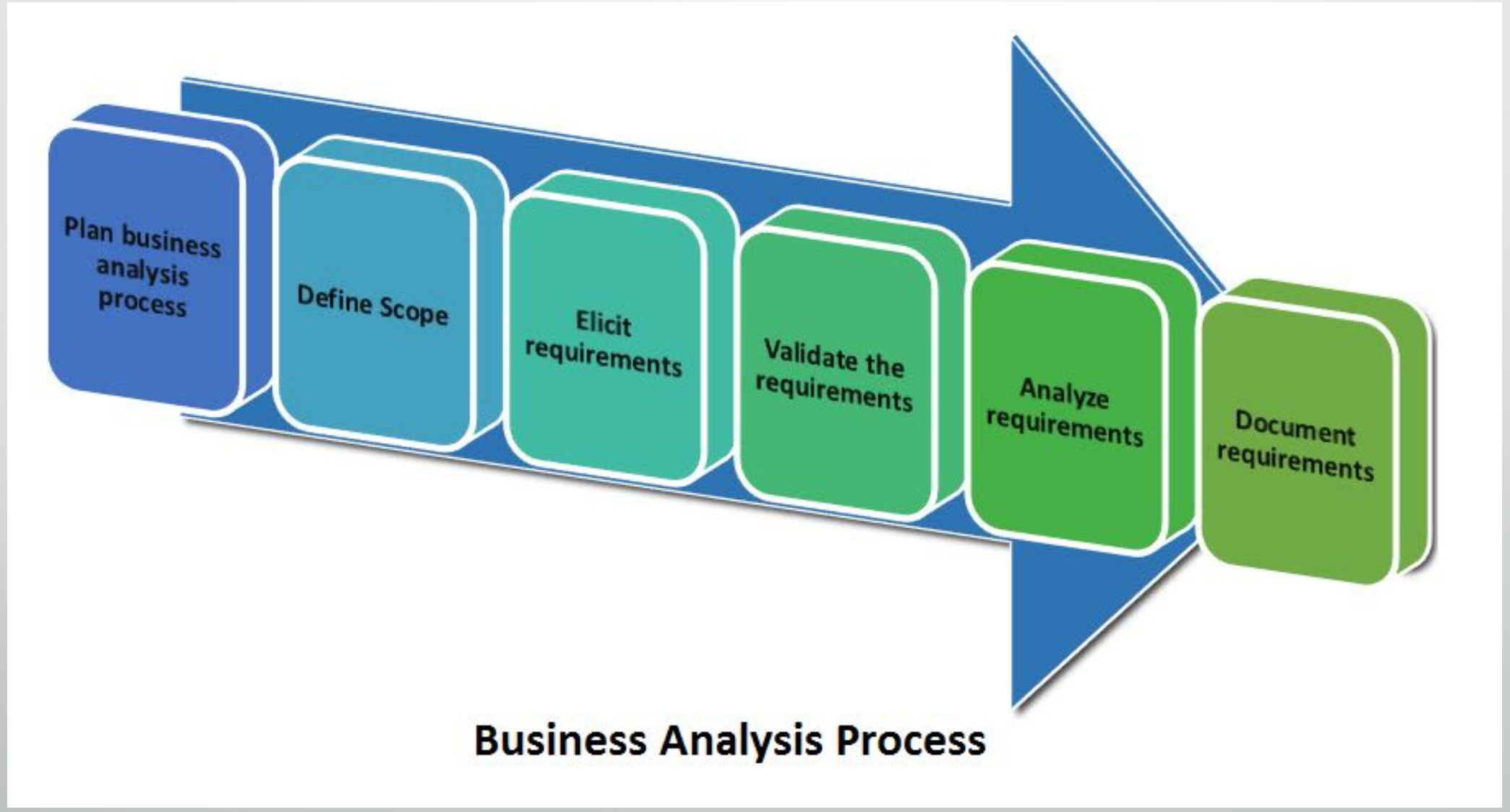
Requirement & other processes



Requirement from various sources

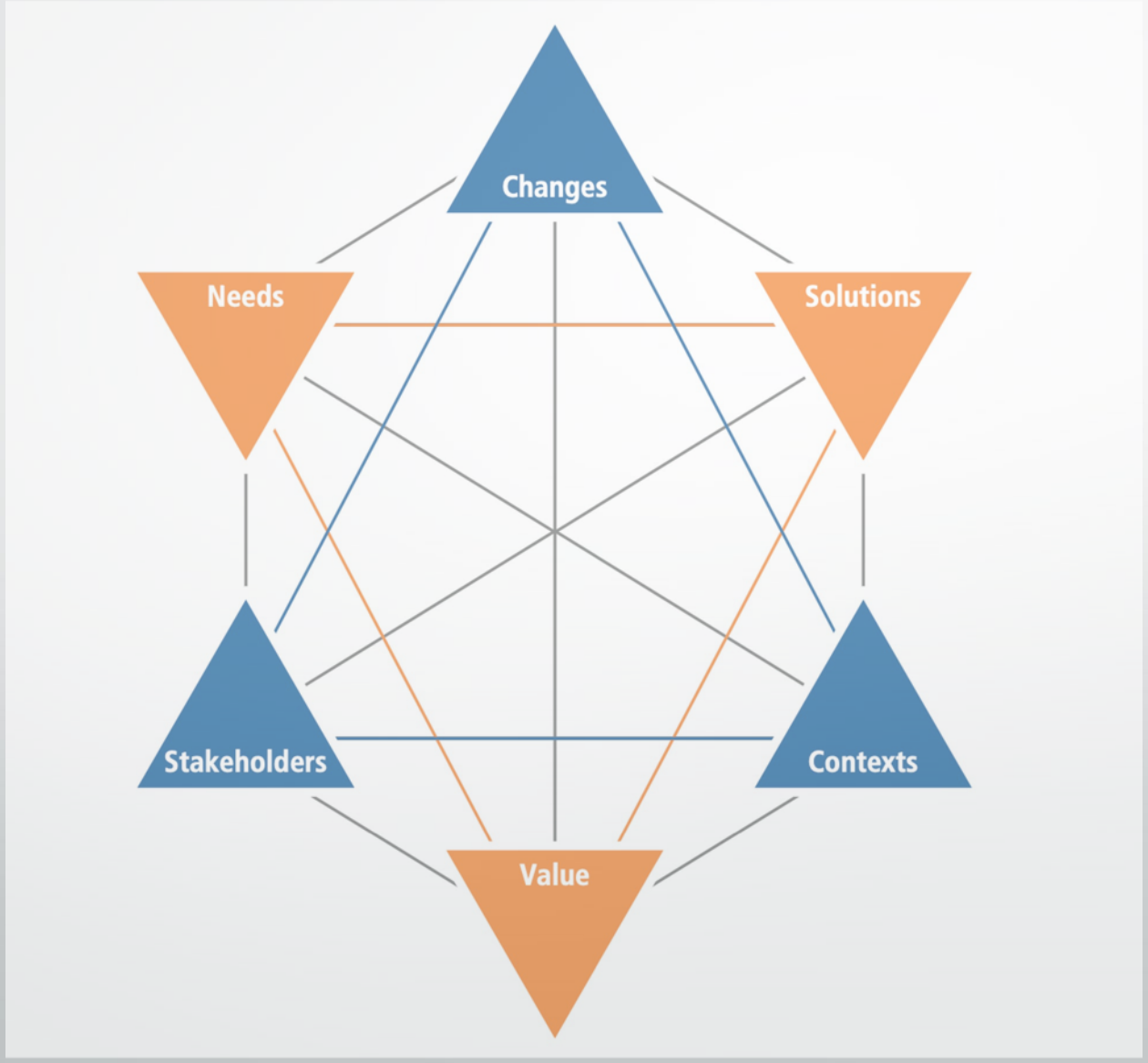


Requirement Development

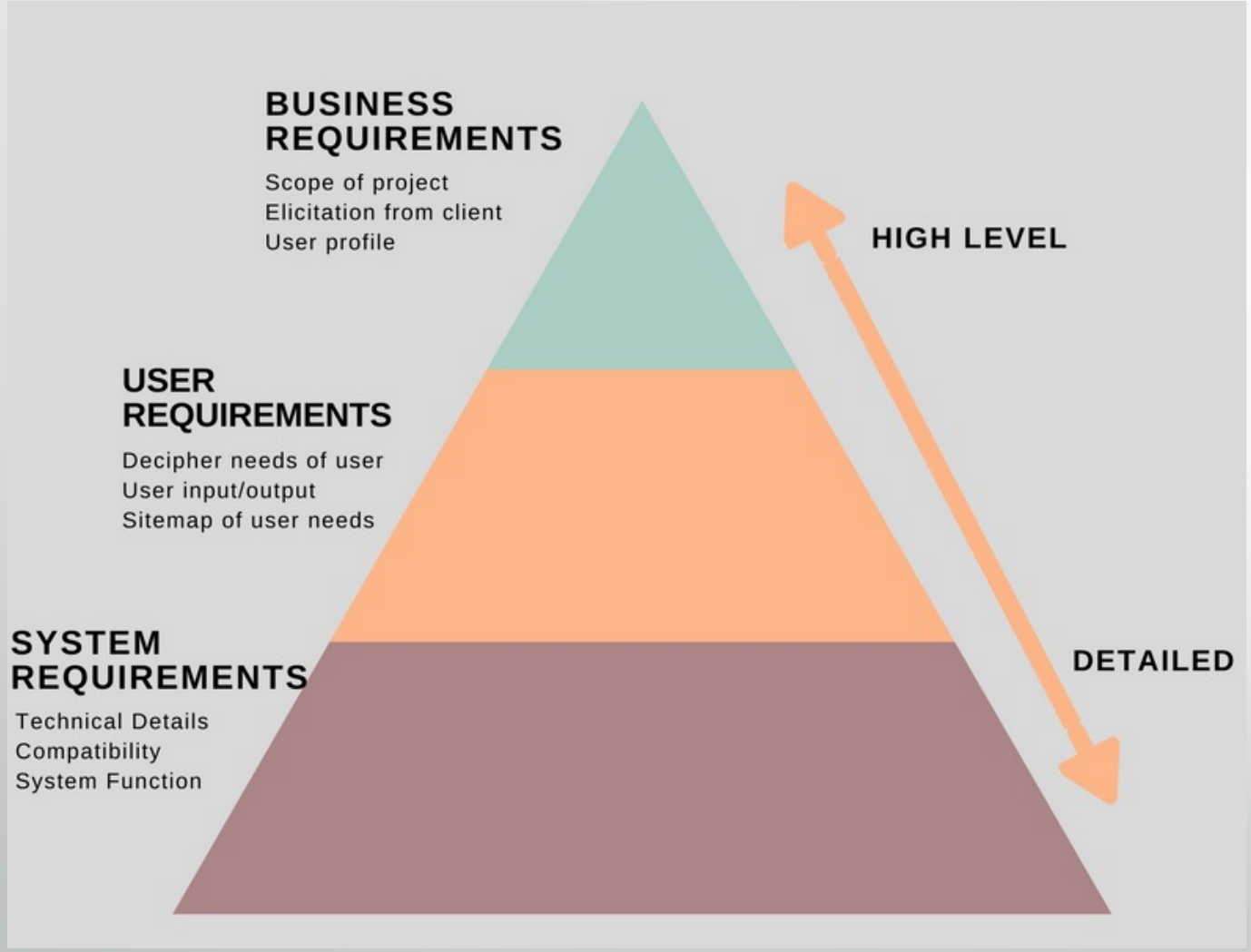


Business Analysis Process

Core Concepts



Requirement Gathering

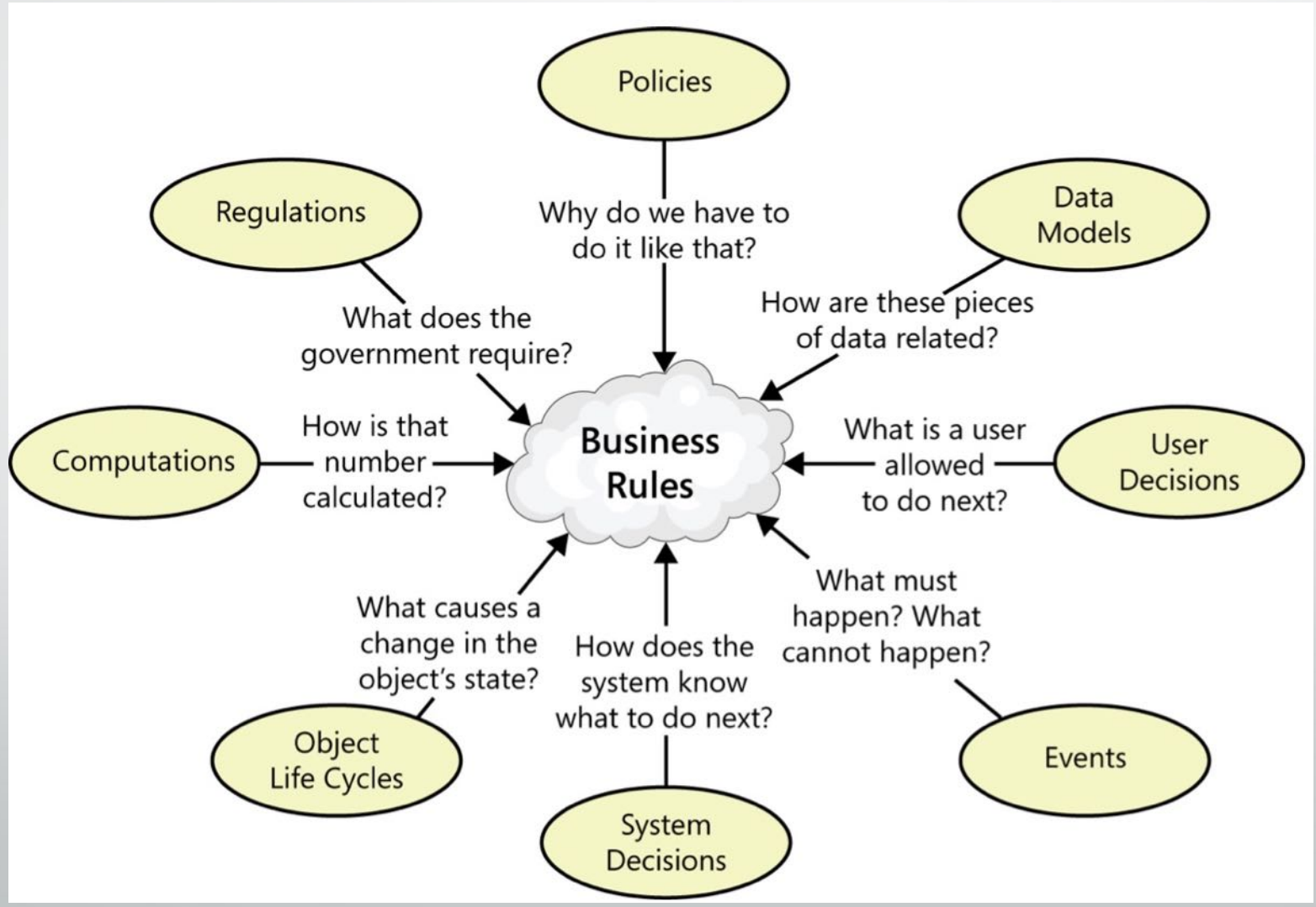


Requirements with views

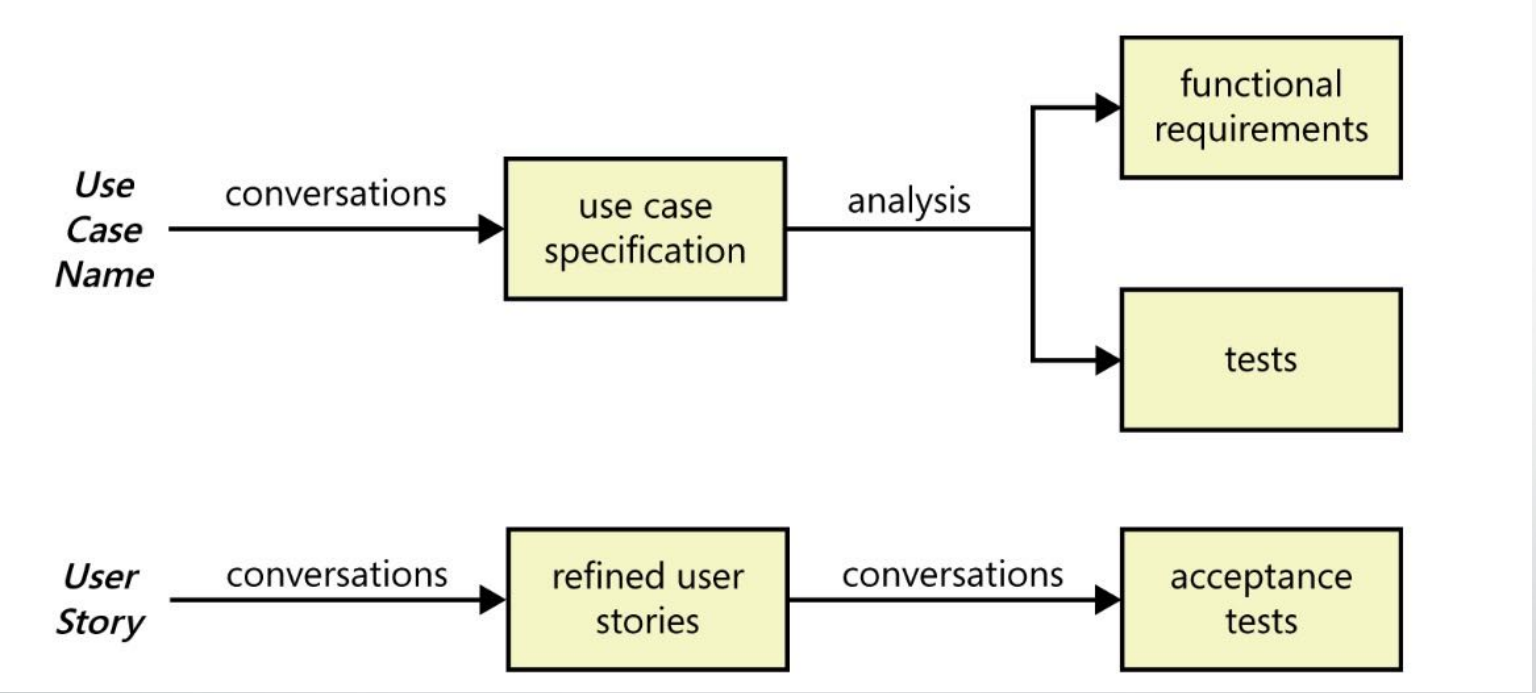
usage-centric: emphasizes understanding and exploring user goals to derive the necessary system functionality

Product-centric: focuses on defining features that you expect will lead to marketplace or business success

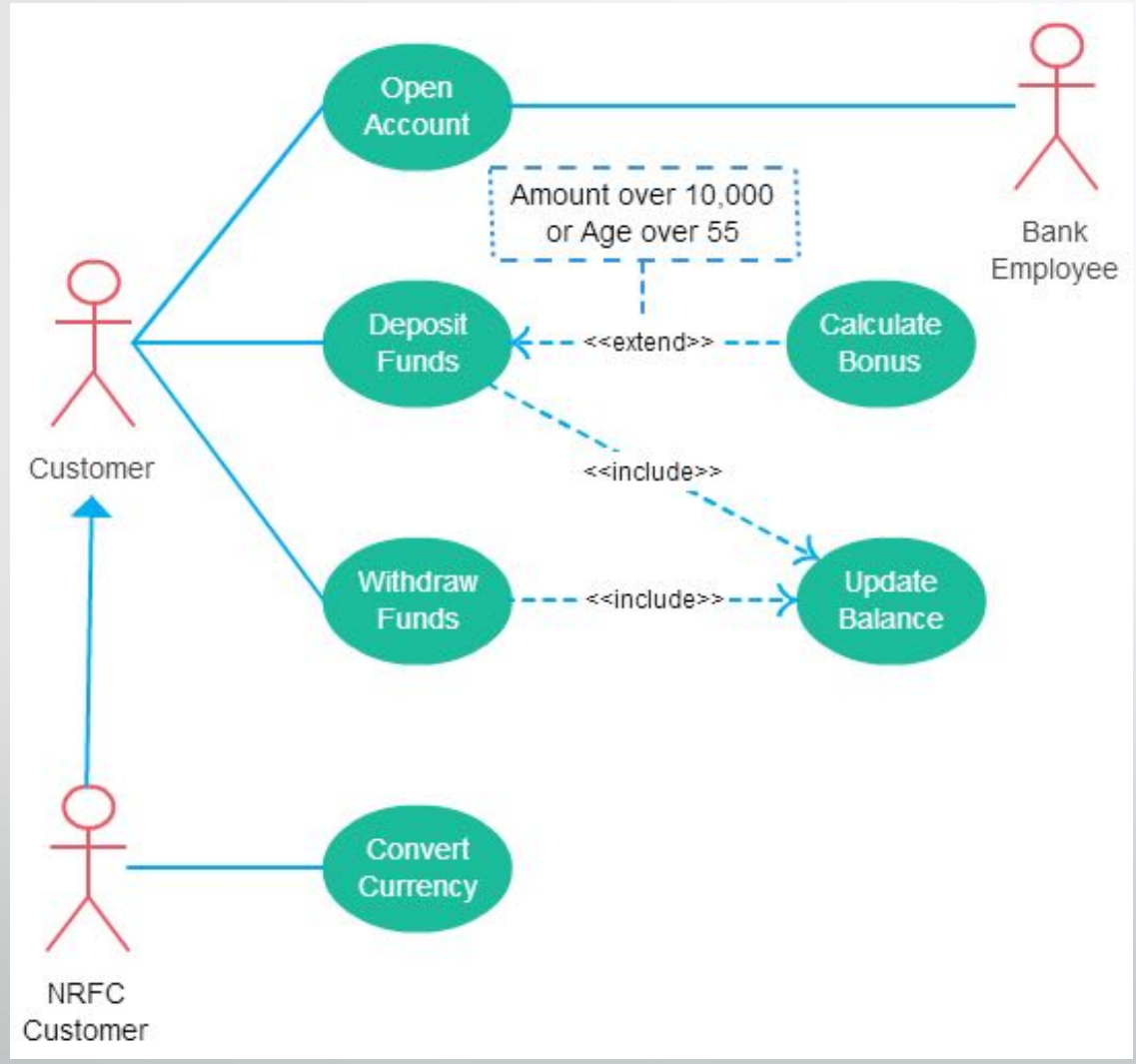
Business rules



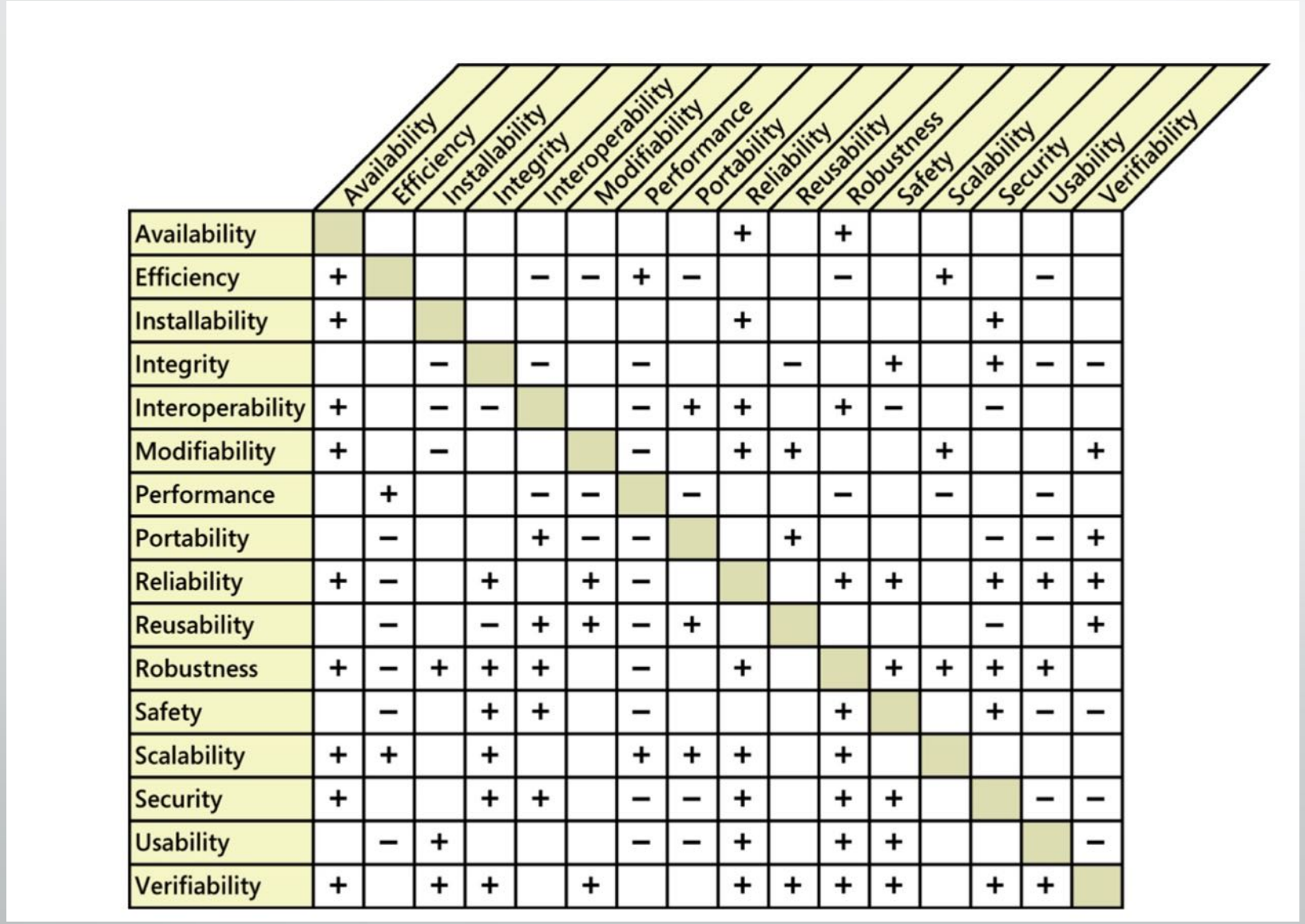
Requirement modelling



Use case

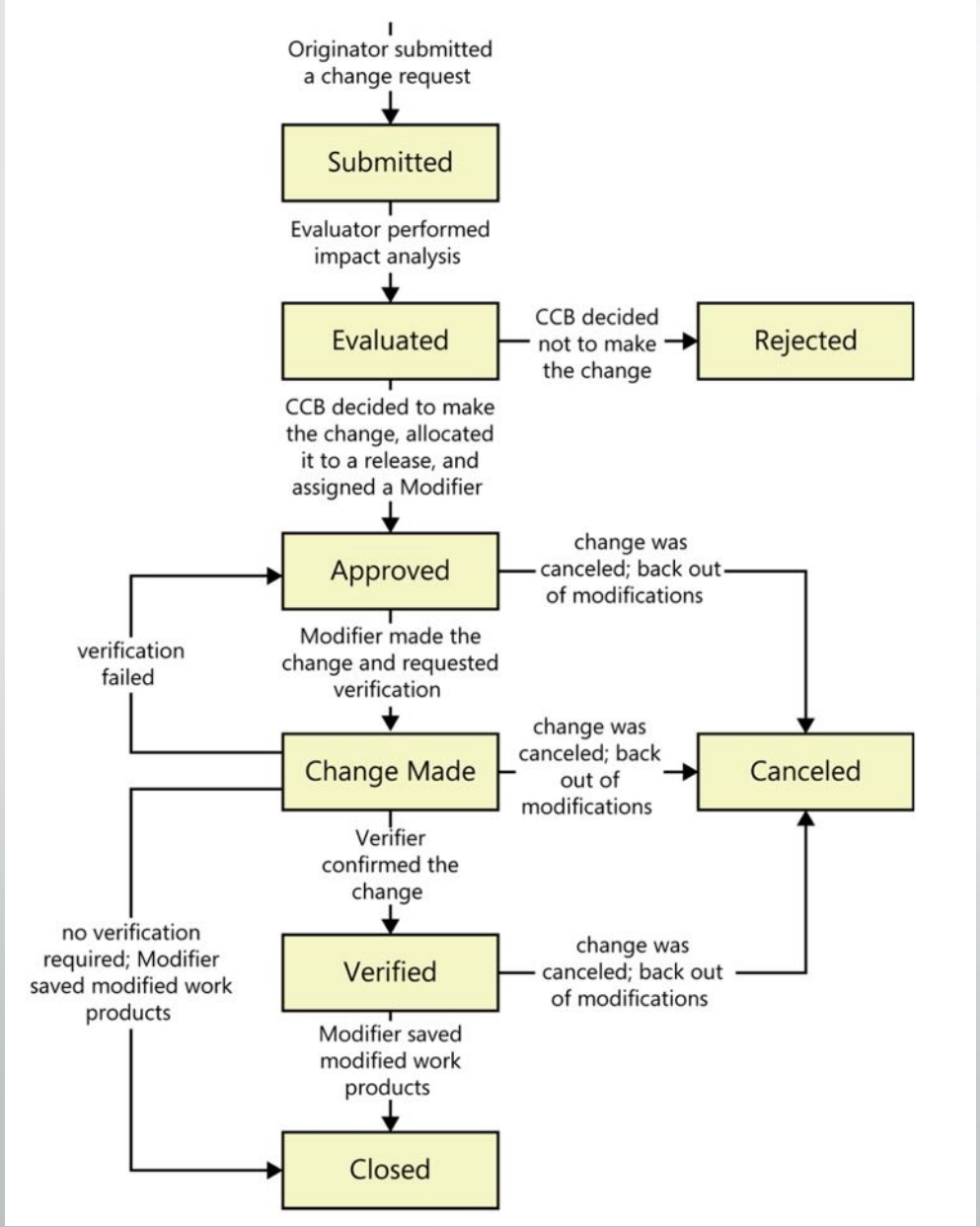

Open Account
Bank Employee
Amount over 10,000 or Age over 55
Deposit Funds
<<extend>>
Calculate Bonus
Customer
<<include>>
Withdraw Funds
<<include>>
Update Balance
Convert Currency
NRFC Customer

Quality Attributes



	Availability	Efficiency	Installability	Integrity	Interoperability	Modifiability	Performance	Portability	Reliability	Reusability	Robustness	Safety	Scalability	Security	Usability	Verifiability
Availability									+		+					
Efficiency	+				−	−	+	−			−		+		−	
Installability	+								+					+		
Integrity			−		−		−			−		+		+	−	−
Interoperability	+		−	−			−	+	+		+	−		−		
Modifiability	+		−				−		+	+			+			+
Performance		+			−	−		−			−		−		−	
Portability		−			+	−	−			+				−	−	+
Reliability	+	−		+		+	−				+	+		+	+	+
Reusability		−		−	+	+	−	+						−		+
Robustness	+	−	+	+	+		−		+			+	+	+	+	
Safety		−		+	+		−				+			+	−	−
Scalability	+	+		+			+	+	+		+					
Security	+			+	+		−	−	+		+	+			−	−
Usability		−	+				−	−	+		+	+				−
Verifiability	+		+	+		+			+	+	+	+		+	+	

Change requests



Entry criteria

Exit criteria

A requirement & baseline

Complete

Correct

Necessary

Feasible

Verifiable

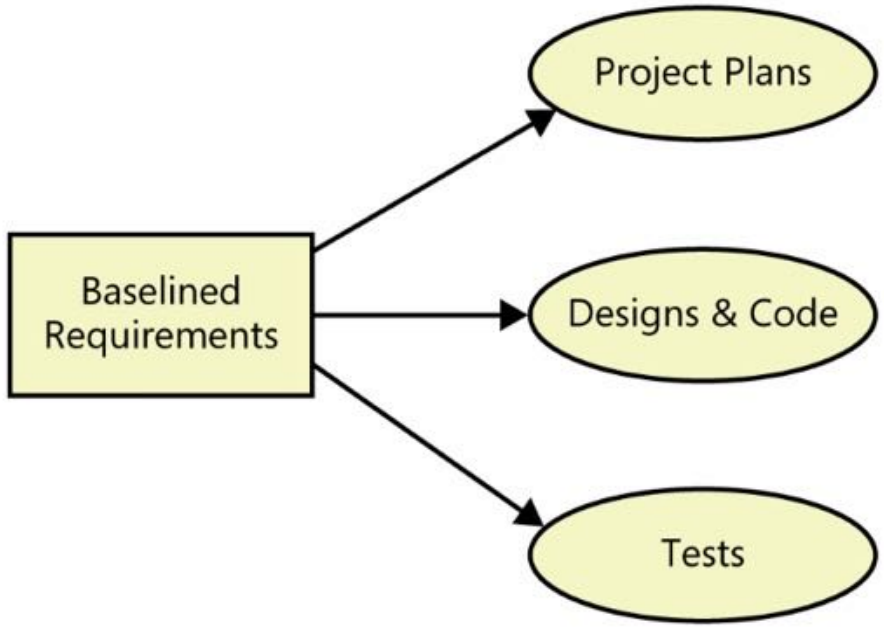
Prioritized

Consistent

Traceable

Unambiguous

Modifiable



- Use requirements to size the project or iteration
- Base estimates on product size
- Update plans as requirements change
- Use requirement priorities to drive iterations

- Have developers review requirements
- Use quality attributes to drive architecture
- Allocate requirements to components
- Trace requirements to designs and code

- Start test design early
- Have users create acceptance tests
- Base system testing on requirements
- Trace requirements to tests

A requirement elicitation

- **Interview**
- **Workshop**
- **Focus group**
- **Observation**
- **Questionnaire**
- **System interface analysis**
- **User interface analysis**
- **Document analysis**

	Interviews	Workshops	Focus groups	Observations	Questionnaires	System interface analysis	User interface analysis	Document analysis
Mass-market software	x		x		x			
Internal corporate software	x	x	x	x		x		x
Replacing existing system	x	x		x		x	x	x
Enhancing existing system	x	x				x	x	x
New application	x	x				x		
Packaged software implementation	x	x		x		x		x
Embedded systems	x	x				x		x
Geographically distributed stakeholders	x	x			x			

Requirement Analysis

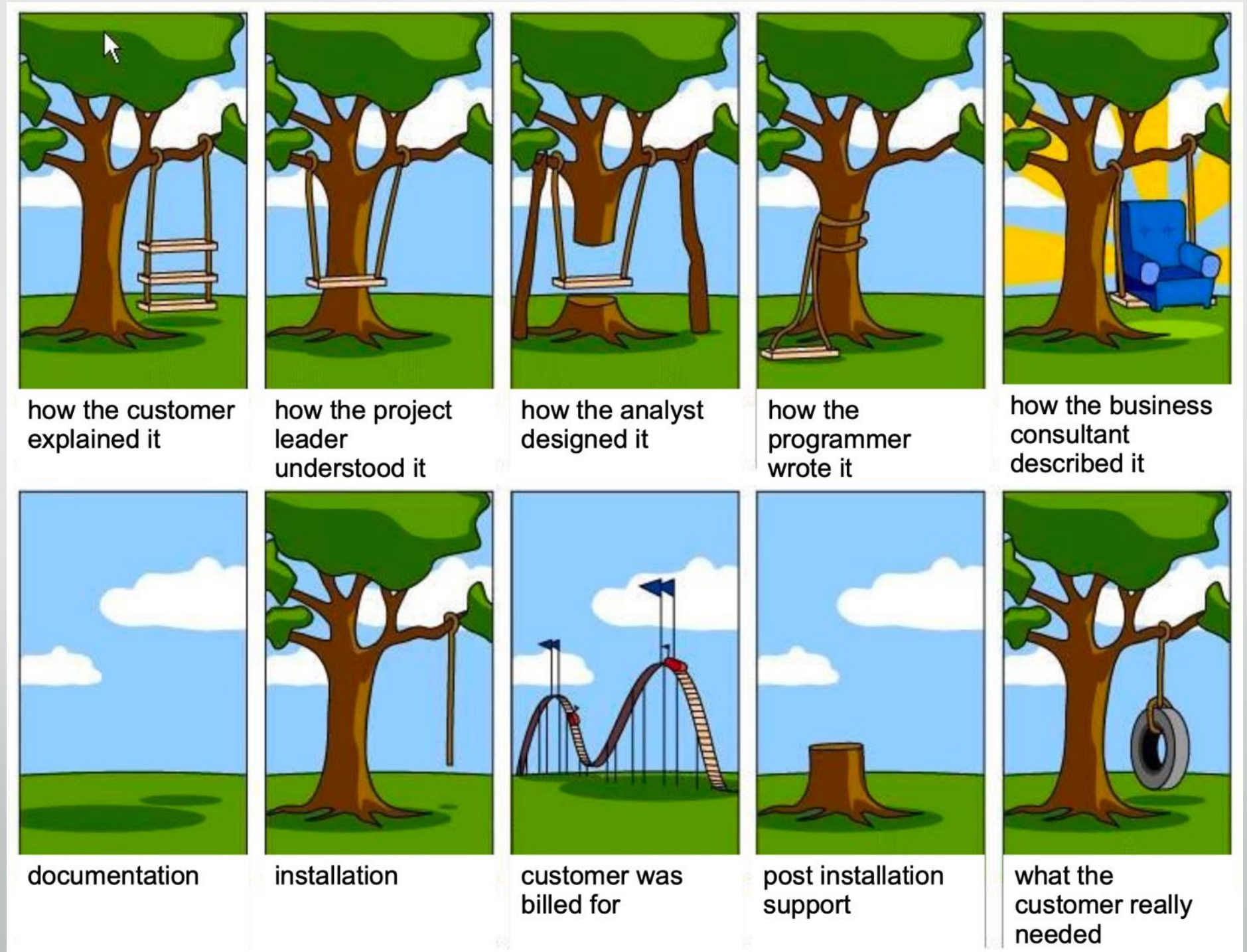
Major activities

- Decomposing **high-level requirement** into appropriate level requirements
- Classify requirement / mapping function – nonfunctional
- Understanding relative importance of quality attributes
- Prioritizing (discuss / negotiate with customer)
- Identify **unnecessary** requirement, gaps (discusses / negotiate with customer)

GAP Analysis

Root cause Analysis

Term	Definition
Assumption	A statement that is believed to be true in the absence of proof or definitive knowledge.
Dependency	As used in requirements specification, a reliance that a project has on a factor, event, or group outside its control.
Architecture	The structure of a system, including any software, hardware, and human components that make up the system, the interfaces and relationships between those components, and the component behaviors that are visible to other components.
External interface requirement	A description of a connection between a software system and a user, another software system, or a hardware device.
User interface (UI)	is the point of human-computer interaction and communication in a device.
Graphical user interface (GUI)	is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.
Software interfaces	Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components.
Hardware interfaces	Describe the characteristics of each interface between the software components and hardware components, if any, of the system.
Communications interfaces	State the requirements for any communication functions the product will use, including email, web browser, network protocols, and electronic forms.


how the customer explained it
how the project leader understood it
how the analyst designed it
how the programmer wrote it
how the business consultant described it
documentation
installation
customer was billed for
post installation support
what the customer really needed

Requirement Specification

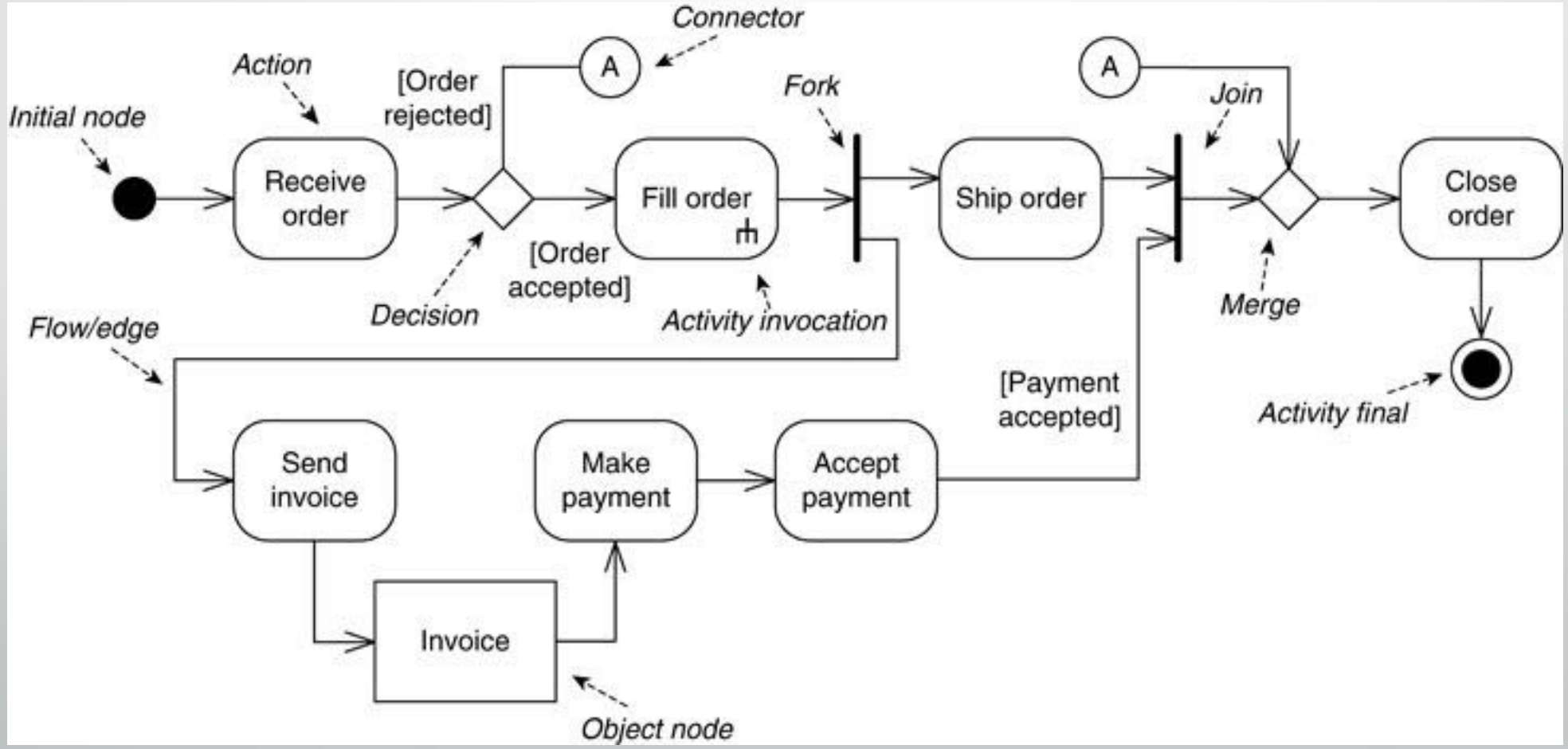
Diagrams

- Ecosystem diagram
- Context diagram (DFD level 0)
- Dialog map
- Decision table
- Decision tree
- Event- response table
- Swimlane diagram
- Data flow diagram
- Flow chart
- State-transition diagram
- State table
- User story

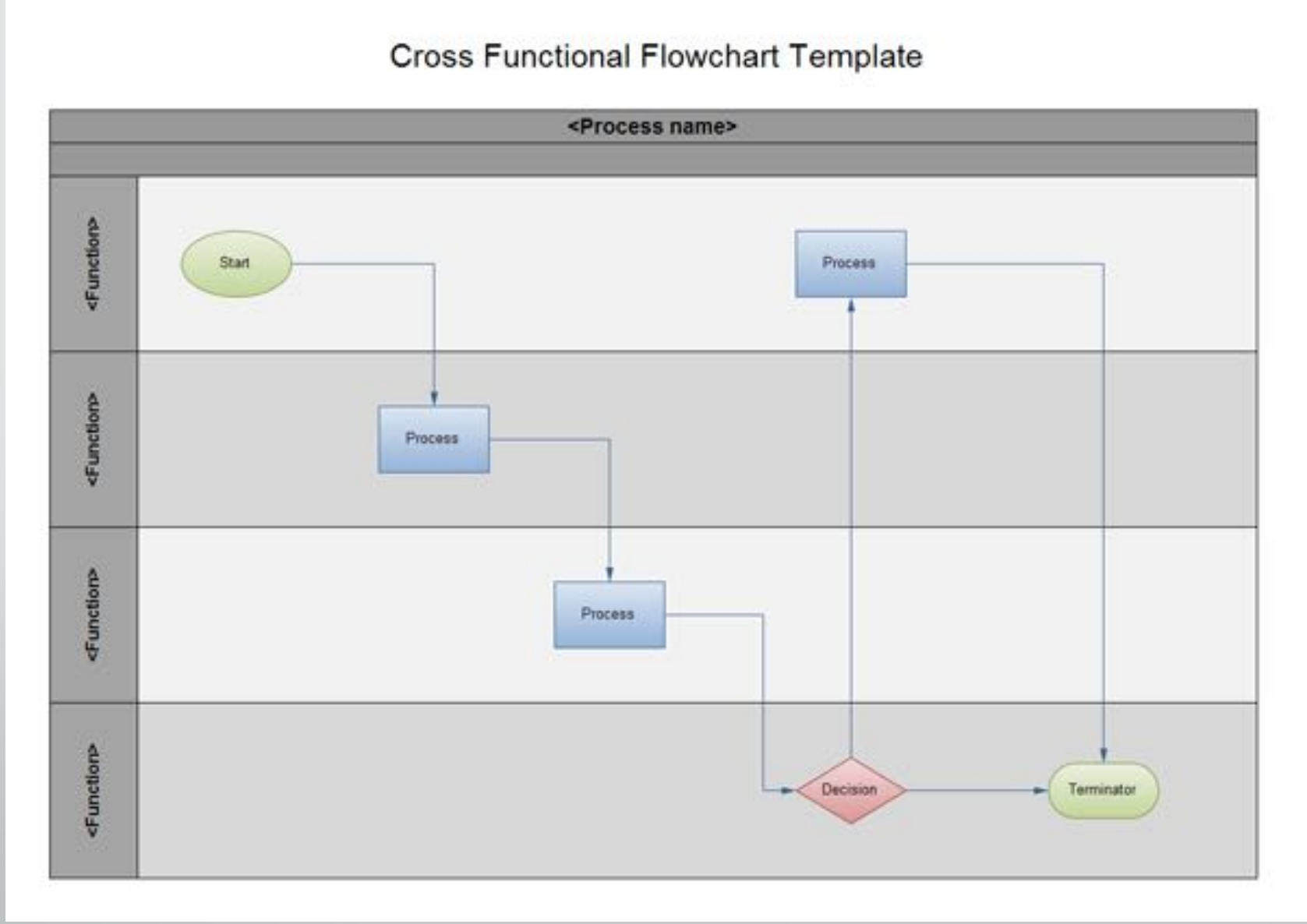
UML

- Activity diagram
- Class diagram
- ERD – entity relationship diagram
- State machine / State diagram
- Use-case diagram

Activity


Connector
A
Action
[Order rejected]
Fork
A
Join
Initial node
Receive order
Fill order
Ship order
Close order
[Order accepted]
Decision
Activity invocation
Merge
Flow/edge
[Payment accepted]
Activity final
Send invoice
Make payment
Accept payment
Invoice
Object node

Swimlane


Cross Functional Flowchart Template
<Process name>
<Function>
<Function>
<Function>
<Function>
Start
Process
Process
Process
Decision
Terminator

Planning

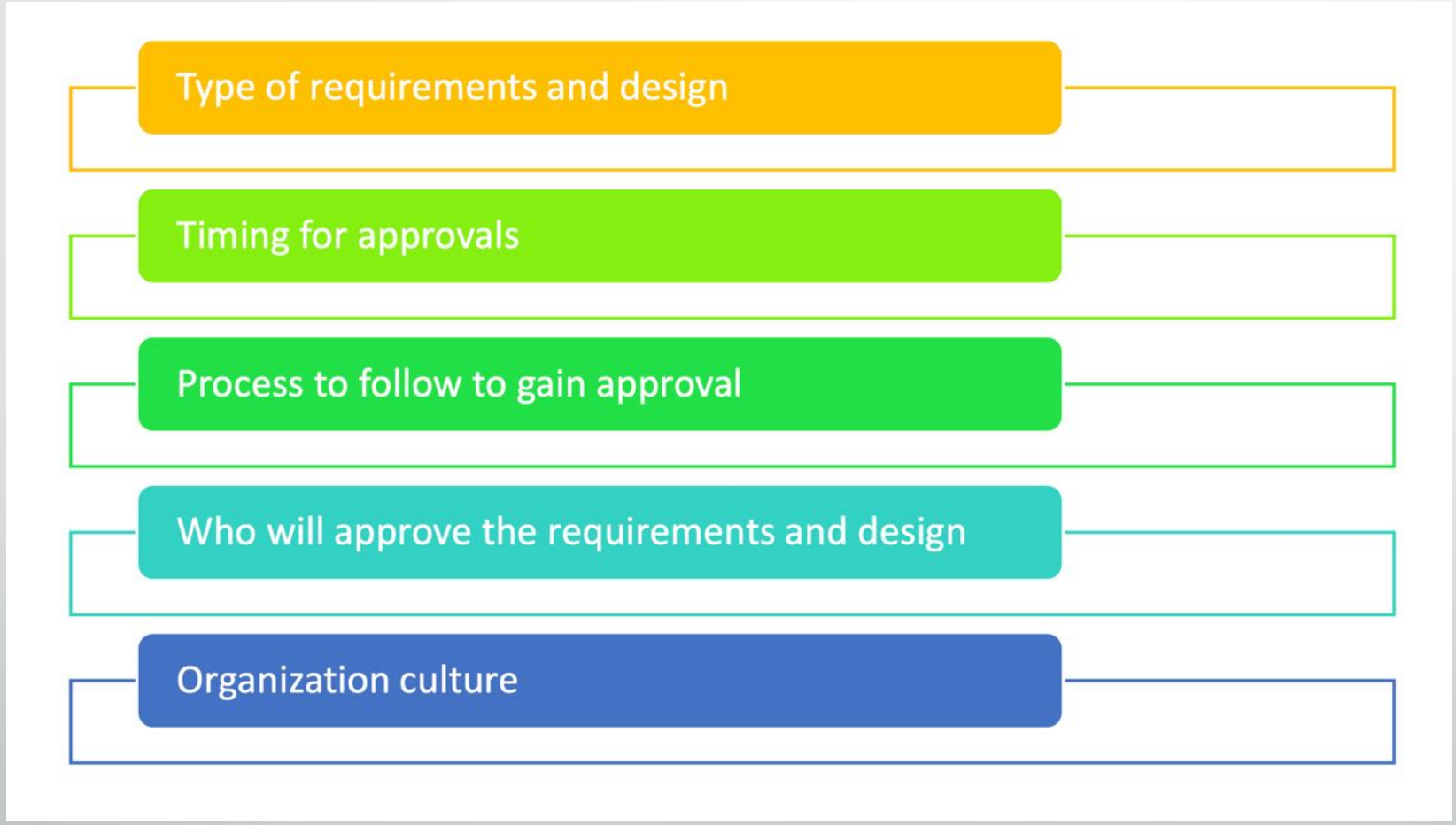
Plan business analysis approach

Plan Stakeholder Engagement

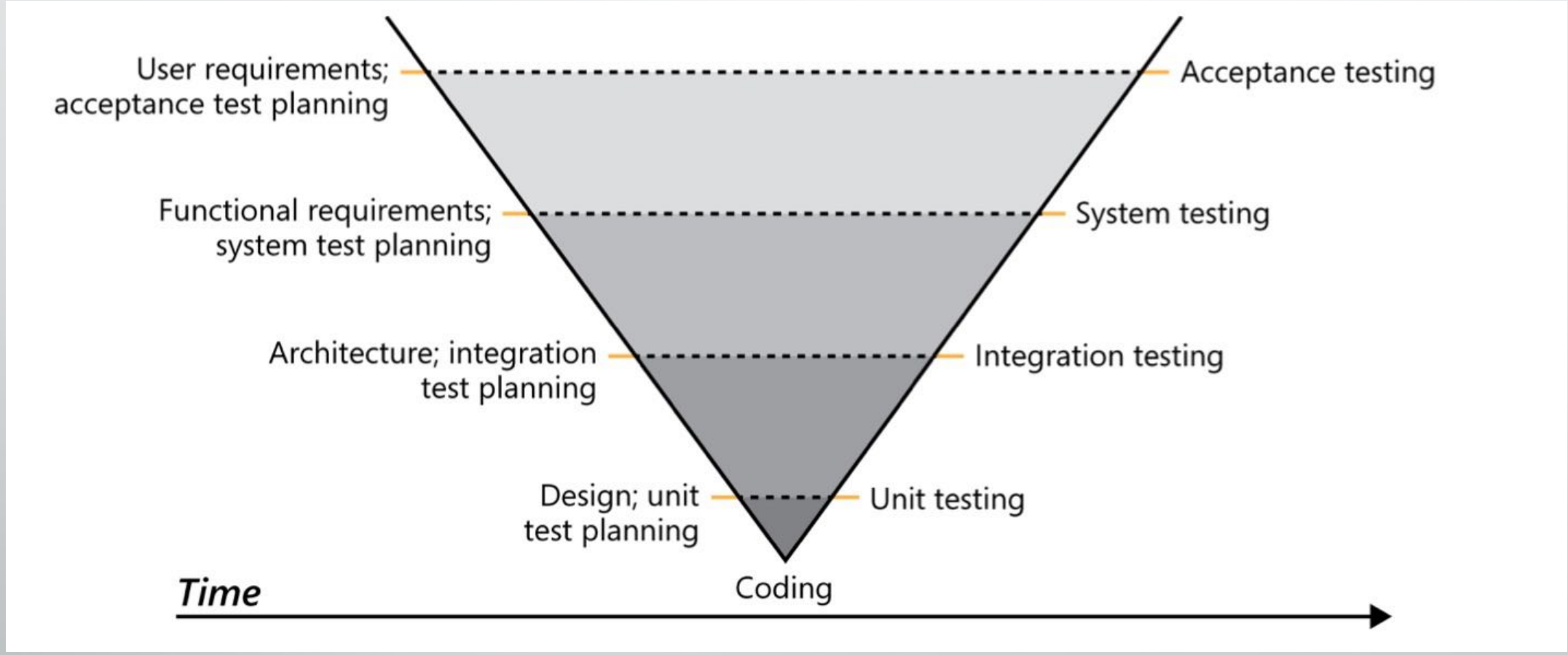
Plan Business Analysis Governance

Plan Business Analysis Information Management

Identify Business Analysis Performance Improvements

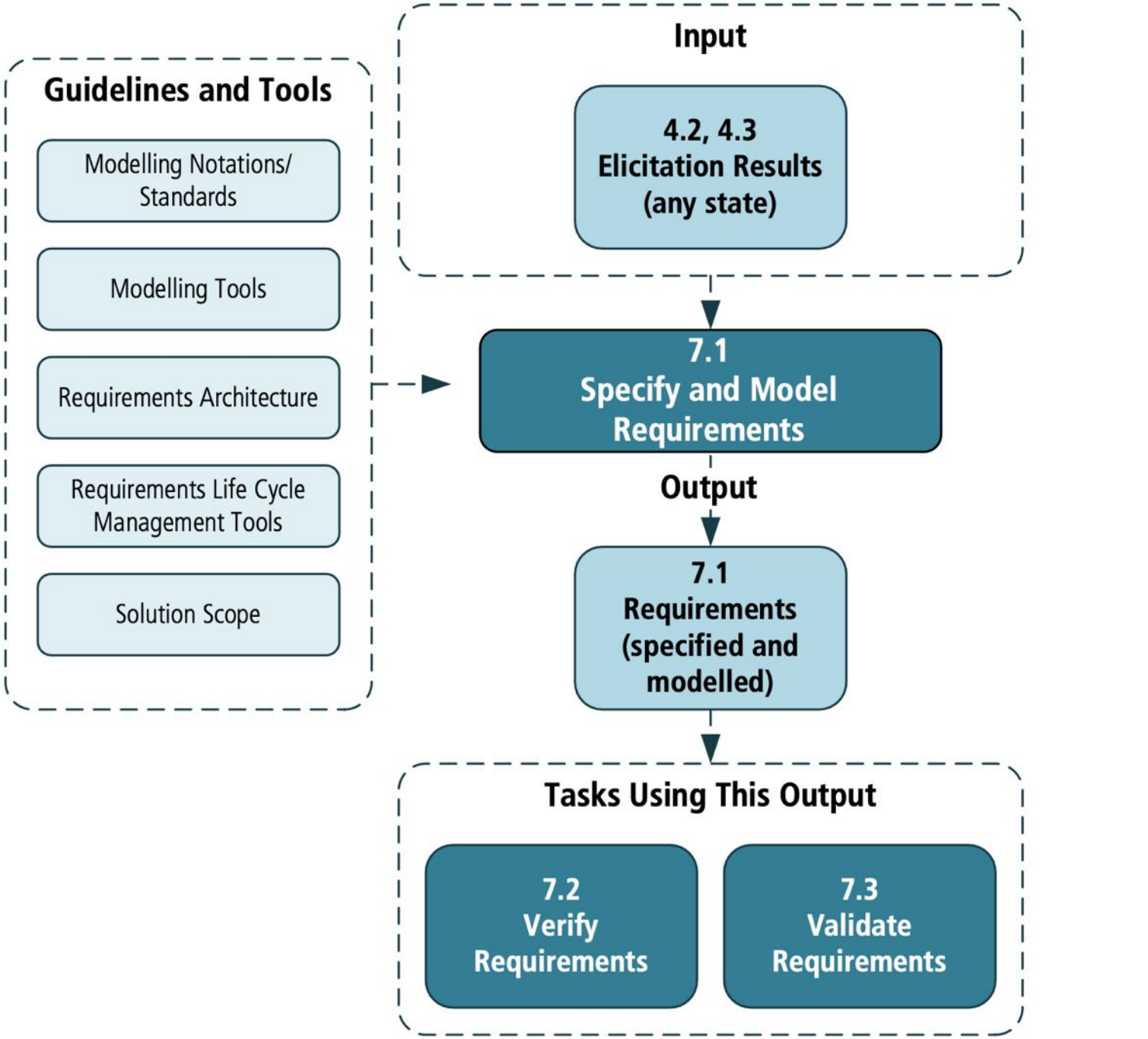

Type of requirements and design
Timing for approvals
Process to follow to gain approval
Who will approve the requirements and design
Organization culture

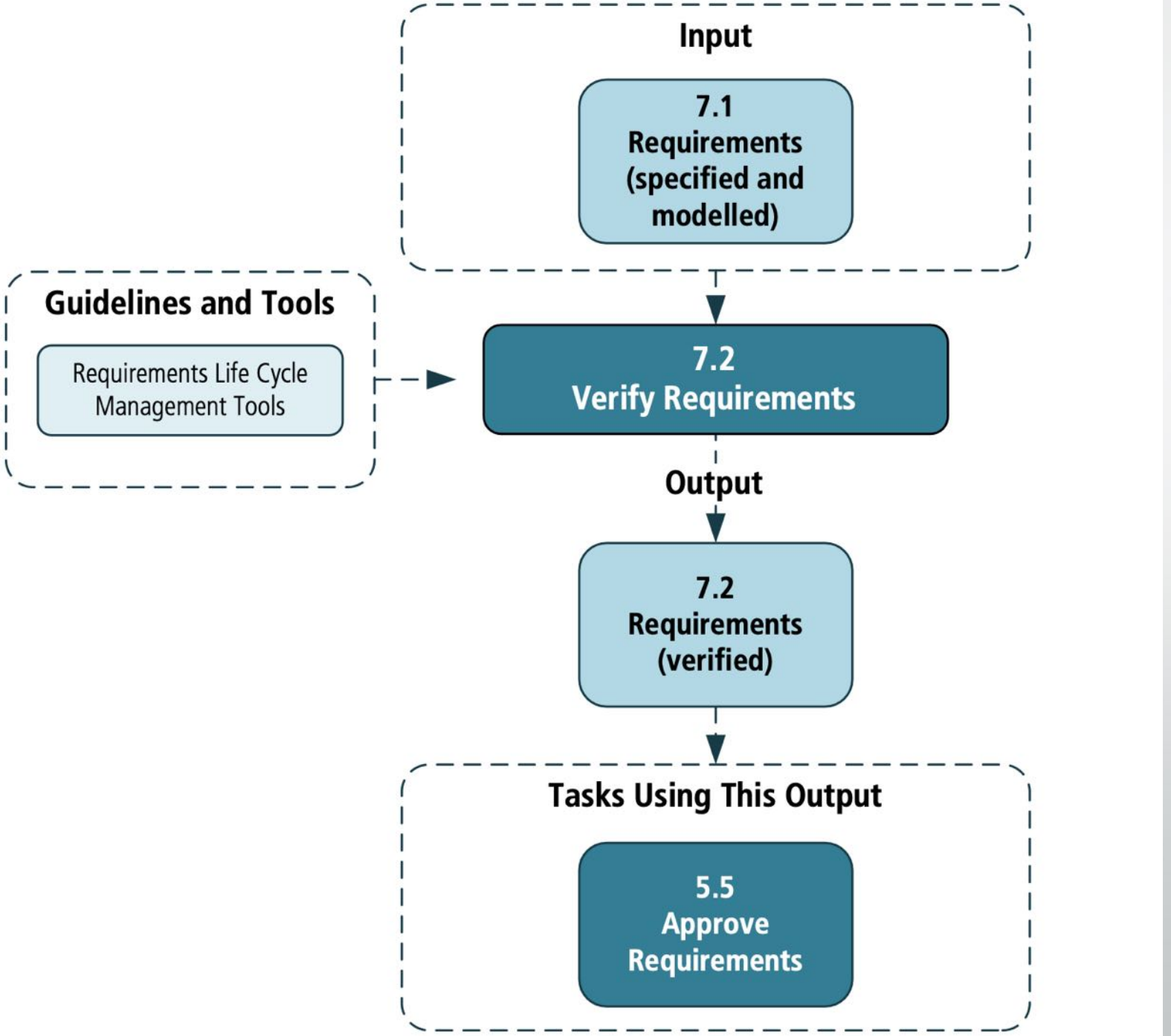
Requirement traceability matrix



Verifying

1. Acceptance criteria
2. Acceptance test
3. validation
4. verification
5. Review
6. Informal review
7. Peer review
8. Peer desk-check
9. Pass-around
10. Walkthrough
11. Formal review
12. Inspection
13. Retrospective

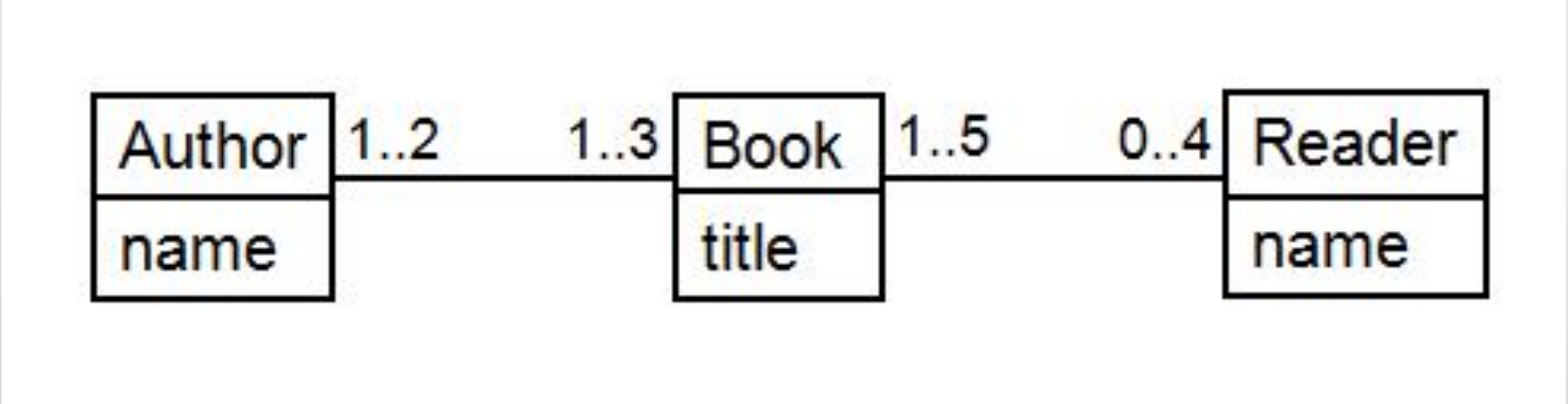

Input
4.2, 4.3
Elicitation Results
(any state)
Guidelines and Tools
Modelling Notations/
Standards
Modelling Tools
Requirements Architecture
Requirements Life Cycle
Management Tools
Solution Scope
7.1
Specify and Model
Requirements
Output
7.1
Requirements
(specified and
modelled)
Tasks Using This Output
7.2
Verify
Requirements
7.3
Validate
Requirements


Input
7.1
Requirements
(specified and
modelled)
Guidelines and Tools
Requirements Life Cycle
Management Tools
7.2
Verify Requirements
Output
7.2
Requirements
(verified)
Tasks Using This Output
5.5
Approve
Requirements

Requirement status

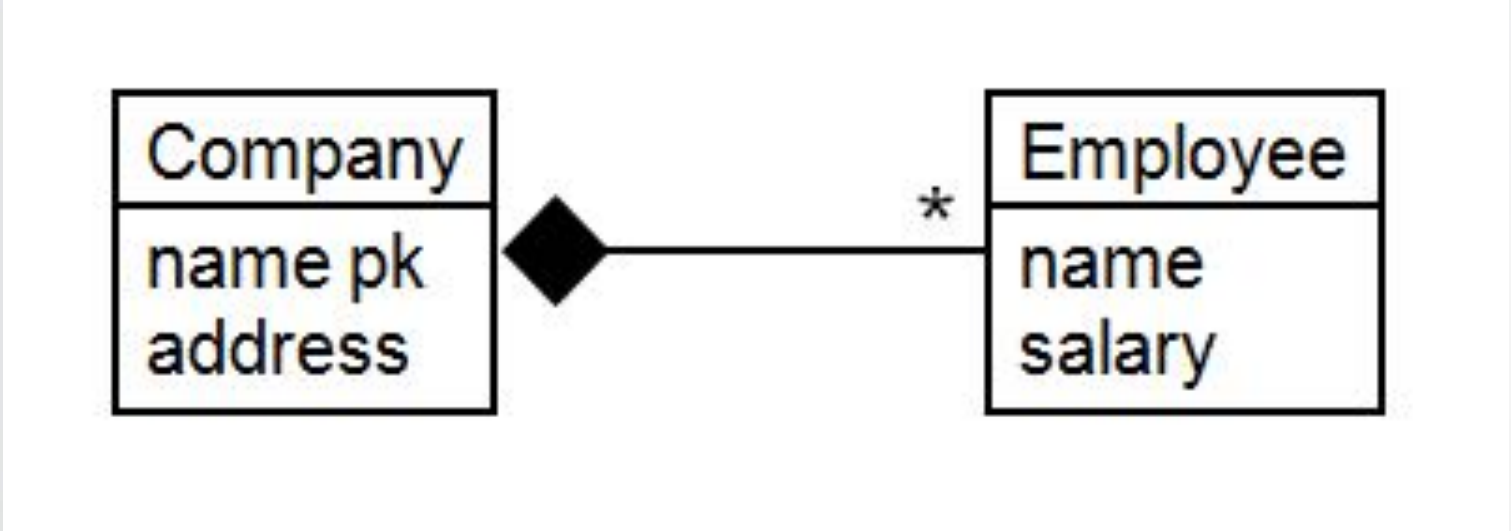
Status	Definition
Proposed	The requirement has been requested by an authorized source.
In Progress	A business analyst is actively working on crafting the requirement.
Drafted	The initial version of the requirement has been written.
Approved	The requirement has been analyzed, its impact on the project has been estimated, and it has been allocated to the baseline for a specific release. The key stakeholders have agreed to incorporate the requirement, and the software development group has committed to implement it.
Implemented	The code that implements the requirement has been designed, written, and unit tested. The requirement has been traced to the pertinent design and code elements. The software that implemented the requirement is now ready for testing, review, or other verification.
Verified	The requirement has satisfied its acceptance criteria, meaning that the correct functioning of the implemented requirement has been confirmed. The requirement has been traced to pertinent tests. It is now considered complete.
Deferred	An approved requirement is now planned for implementation in a later release.
Deleted	An approved requirement has been removed from the baseline. Include an explanation of why and by whom the decision was made to delete it.
Rejected	The requirement was proposed but was never approved and is not planned for implementation in any upcoming release. Include an explanation of why and by whom the decision was made to reject it.

Practice



(a) If there are 6 authors, what's the minimum and maximum number of books? What's the minimum and maximum number of readers?
(b) If there are 6 readers, what's the minimum and maxiumum number of books? What's the minumum and maximum number of authors?

Practice



(a) No two companies can have the same name
(b) No two employees can have the same name
(c) No two companies can be at the same address
(d) No two employees can work at the same address
(e) Each employee works for at least one company
(f) No employees work for more than one company
(g) Each company has at least one employee
(h) Two employees with the same name cannot work for the same company
(i) Two employees with the same name cannot work for different companies

Practice

Consider a tiny social network containing high school students and their "crushes" (desired romantic relationships). Each student may have a crush on at most one other student, and associated with each crush is the length of time the crush has been going on. Students have a name and a grade, and names are unique. Draw a UML diagram that models this information. Make sure to capture the asymmetry and multiplicity of the crush relationship.

Practice

Consider a class Book with four subclasses: Anthology, Fiction, Children, and Nonfiction. Is the subclassing relationship overlapping or disjoint (exclusive)? Is it complete or incomplete (partial)?

Practice

Consider a logger;
- Can configure file format, file name, database,..
- Can log into DB or file
- When any activity updated in the system, that action must be logged
- a log record will have several types: info, warning, error
- a log writer will have several levels: 0, 1, 2
---- level 0: only error will be recorded
---- level 1: only error and warning will be recorded
---- level 2: all log will be recorded

➔ Draw class diagram
➔ Draw state transition diagram

Practice

Consider a bike rental system.

There is a bicycle rental system. Clients can rent several bicycles for a maximum of 10 days, and if they do not return rented bikes on time, they will have to pay a penalty of 10 USD per day per bike. There are various bicycle models to choose and there are stores to pick bikes up. A reservation must have enough info for renting. Pickup day must go before the return day.

➔ Draw ERD
➔ Draw state transition diagram for a bike and reservation

Practice

Design ATM system
- cash withdrawal
- view statement (5 recent transactions)
- check balance
- check amount (% 50,000; <=5,000,000) ; number of cash (<=35 pieces)
- cash type in ATM: 10,000; 50,000; 100,000 ; 200,000; 500,000
- check account number, account amount
- ATM can count and dispense money
- ATM can be out of service

->> use case diagram / ERD / Business rules for cash withdrawal / activity diagram of cash withdrawal / state transition diagram of cash withdrawal

Practice

QUIZ 1	Which one of the following business analysis techniques is used when prioritizing requirements?

a. Decision Analysis

b. Item Tracking

c. Brainstorming

d. Workshops

QUIZ 2	What is Double Triangular Distribution? If most likely is 9, best case is 5 and worst case estimate is 11.

a. 8

b. 8.66

c. 8.33

d. 1

Practice

QUIZ 3	Which one of the following is not a basis for prioritization?

a. Value
b. Penalty
c. Stability
d. Necessity

QUIZ 4	Which one of the following is an input to the approve requirements task?

a. Requirements (Verified)
b. Requirements (Communicated)
c. Requirements (Prioritized)
d. Requirements (Validated)

Practice

There were conflicts among the stakeholders on the prioritization process.

Key stakeholders were not invited for the prioritization meeting.

The out of scope requirements were allocated to the solution component.

Unable to implement one of the approved requirements.

Implementation has taken more time.

ACTIVITIES

The key stakeholder has decided to design, develop, and implement the solution incrementally.

The high level scope and release plan are in place.

Requirements are progressively elaborated.

Requirements that can be used in future projects have been identified.

	Questions	Response
1	In the given case study, what is the approach for business analysis?	○ Iterative Approach ○ Incremental Approach ○ Predictive ○ Adaptive
2	What should Paul do when he identifies a couple of requirements, which can be used in future projects?	○ Hold for the next projects ○ Label and store for reusability ○ Out of scope as it is general requirement ○ None of the above
3	What can be the reason for the requirements not in scope getting approved and allocated to a solution component for implementation?	○ Change control process was not effectively implemented ○ Impact Analysis was not performed ○ Missing traceability ○ Stakeholder urgency

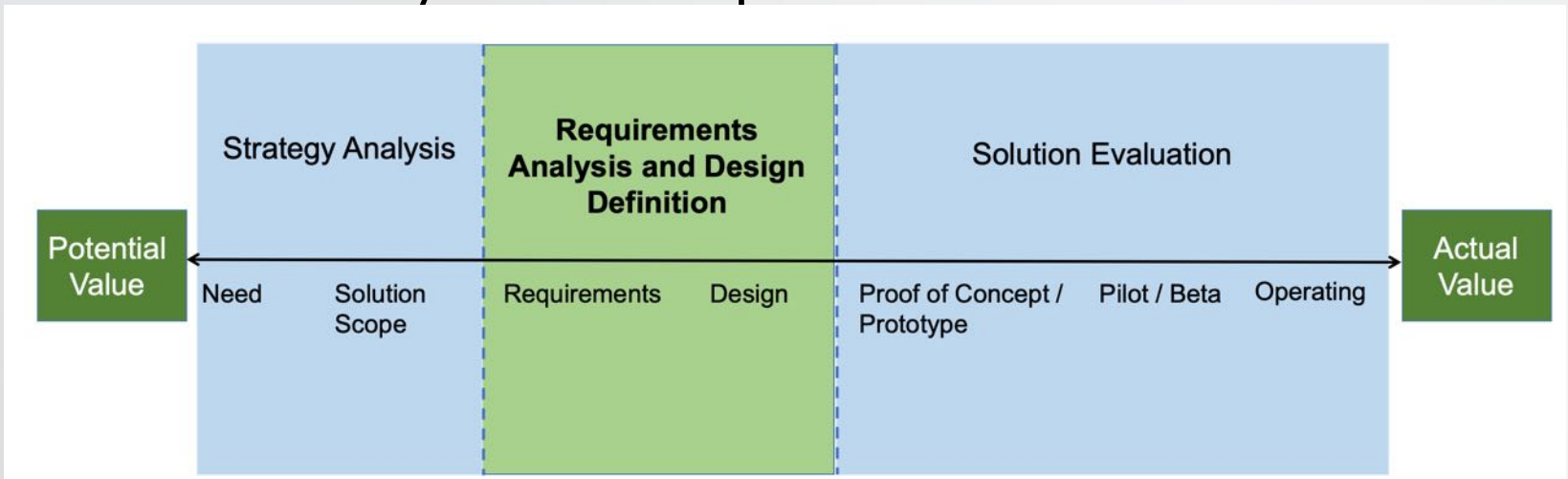
	Questions	Response
4	What can be the reason for missing to invite key stakeholders for the prioritization meeting, when they are required for providing approvals?	○ Forgot to invite ○ Prioritization approach was not adequately defined in the business analysis approach ○ Prioritization approach was not adequately defined in the business analysis governance approach ○ Prioritization approach was not adequately defined in the business analysis information management approach
5	When requirements are prioritized based on only value, what flaw does the approach have?	○ No impact ○ Must prioritize high value requirements ○ Missed considering relationship with other requirements ○ None of the above

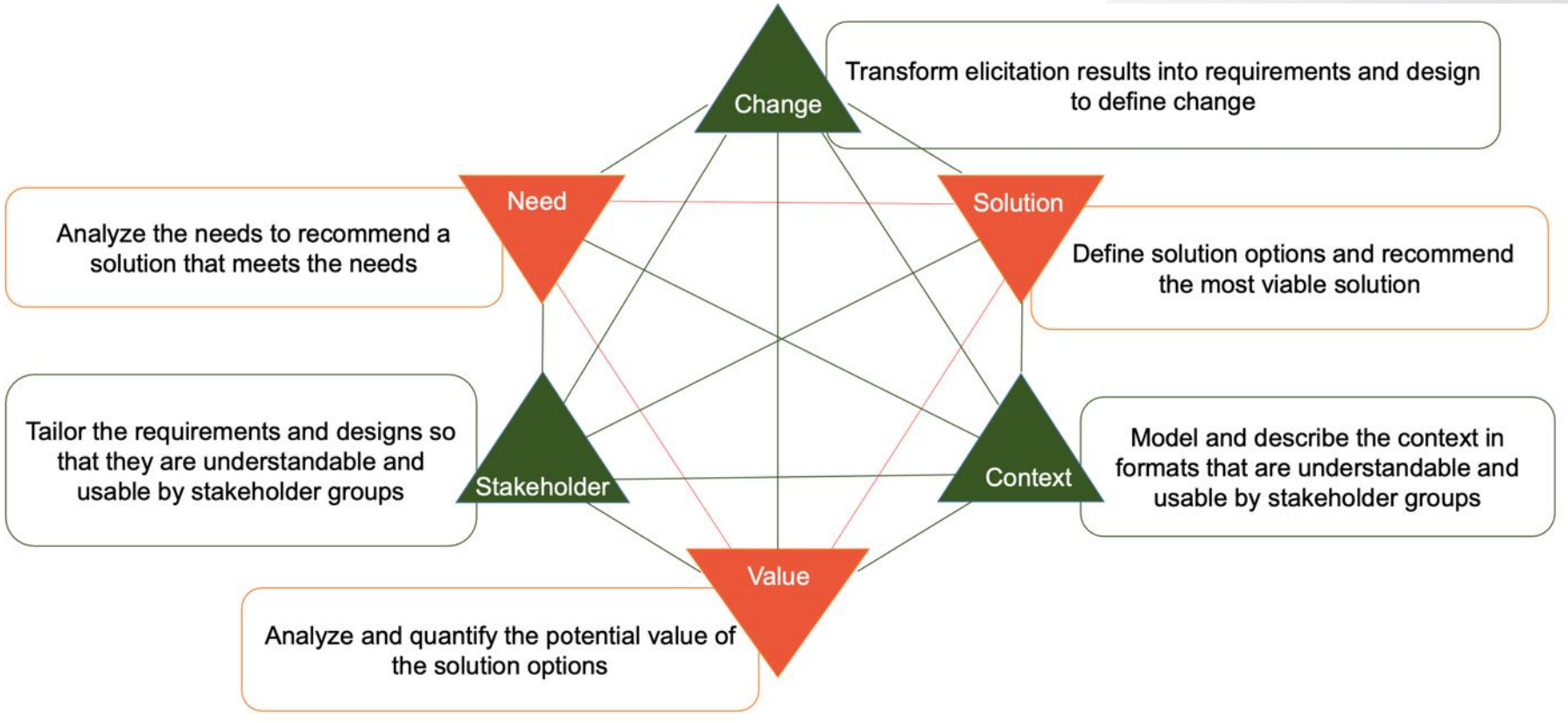
CBAP® CERTIFICATION

Criteria	Requirements for CBAP®® Certification
Work Experience	Minimum 7500 Hours of Business Analysis experience in the last 10 years
Knowledge Area Expertise	Minimum 900 hours in four of the six knowledge areas
Professional Development	35 Hours in the last 4 Years
References	2
Signed Code of Conduct	Yes

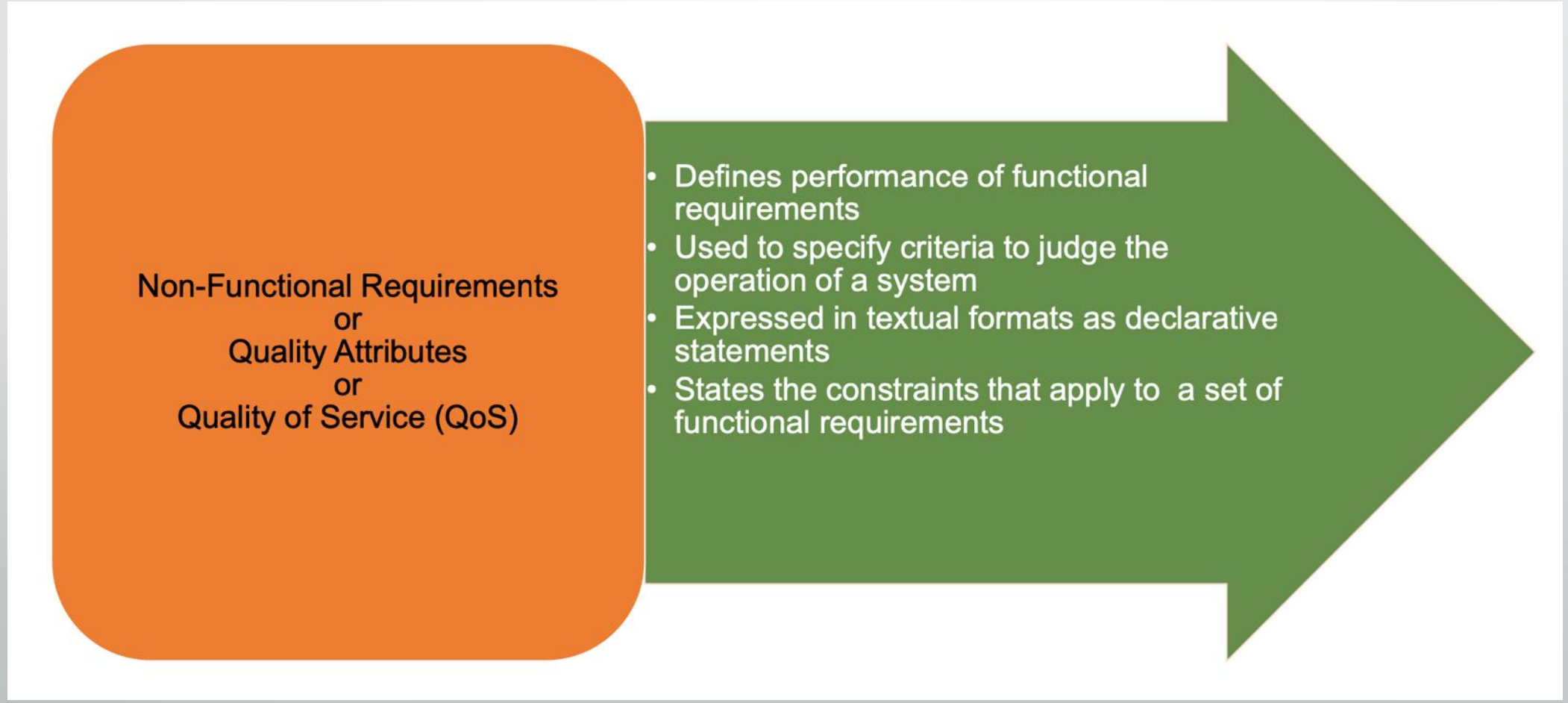
Domain	Percentage of Questions
Business Analysis Planning and Monitoring	14%
Elicitation and Collaboration	12%
Requirements Life Cycle Management	15%
Strategy Analysis	15%
Requirements Analysis and Design Definition	30%
Solution Evaluation	14%

Business analysis value spectrum

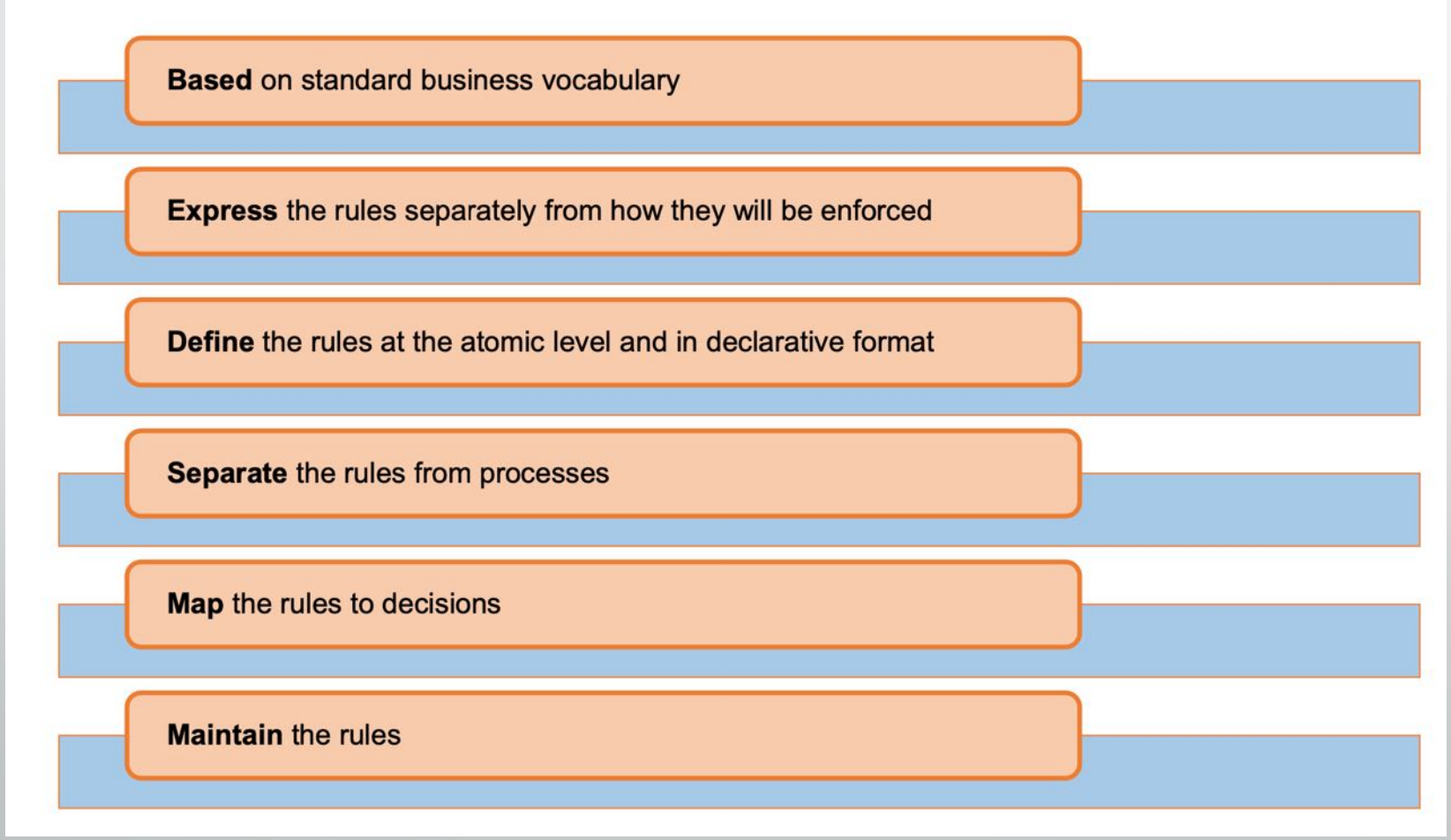




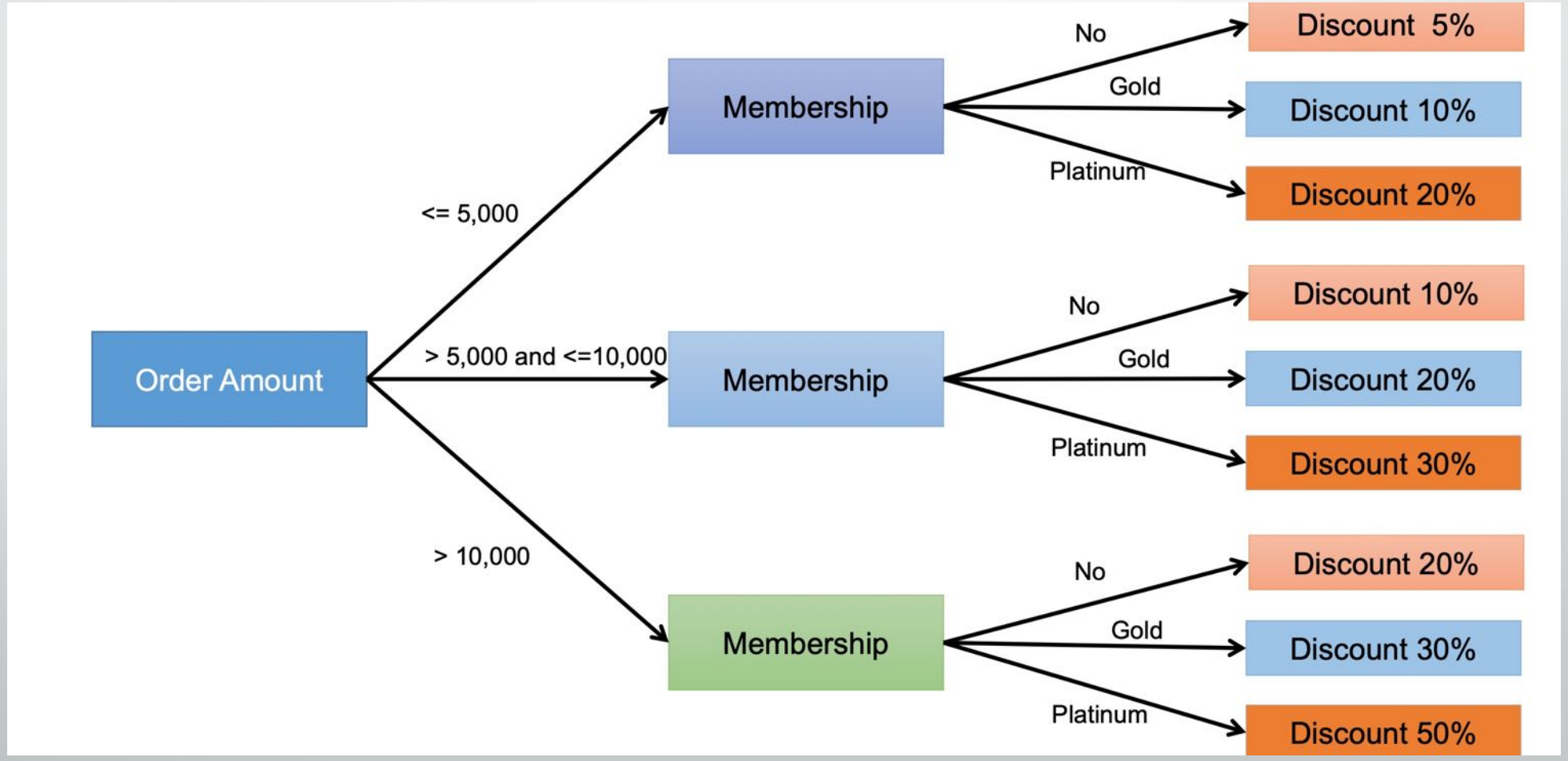
NON-FUNTIONAL REQ. ANALYSIS



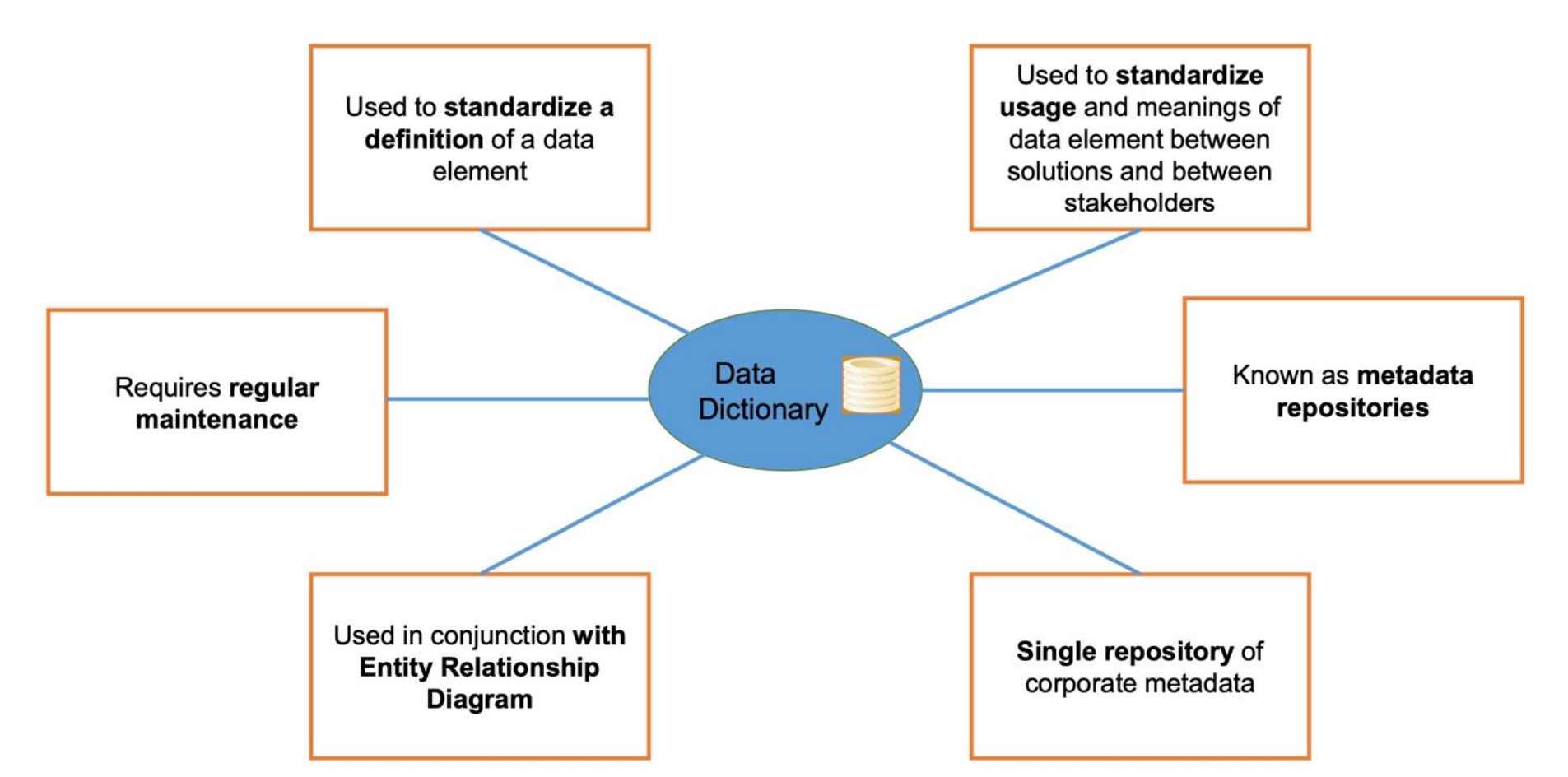
BUSINESS RULES ANALYSIS



DECISION TREE



DATA DICTIONARY



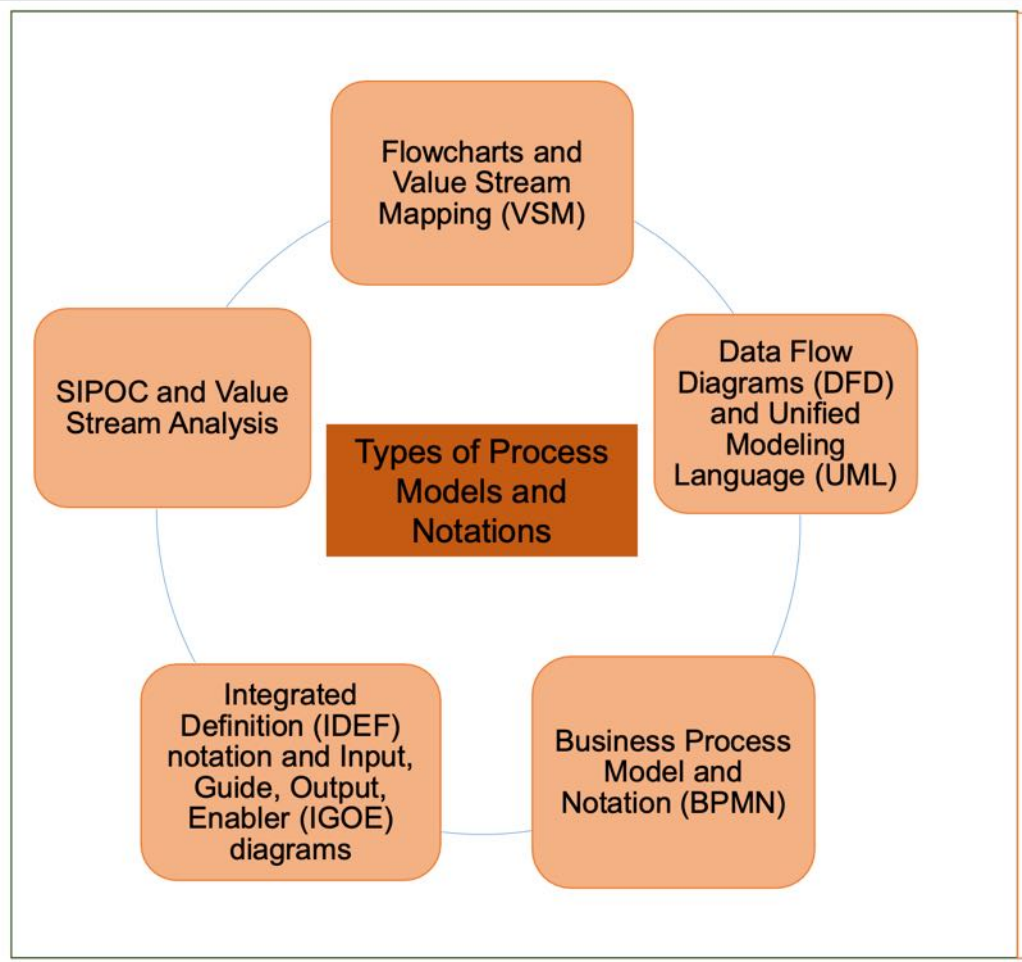
LOGICAL DATA MODELING

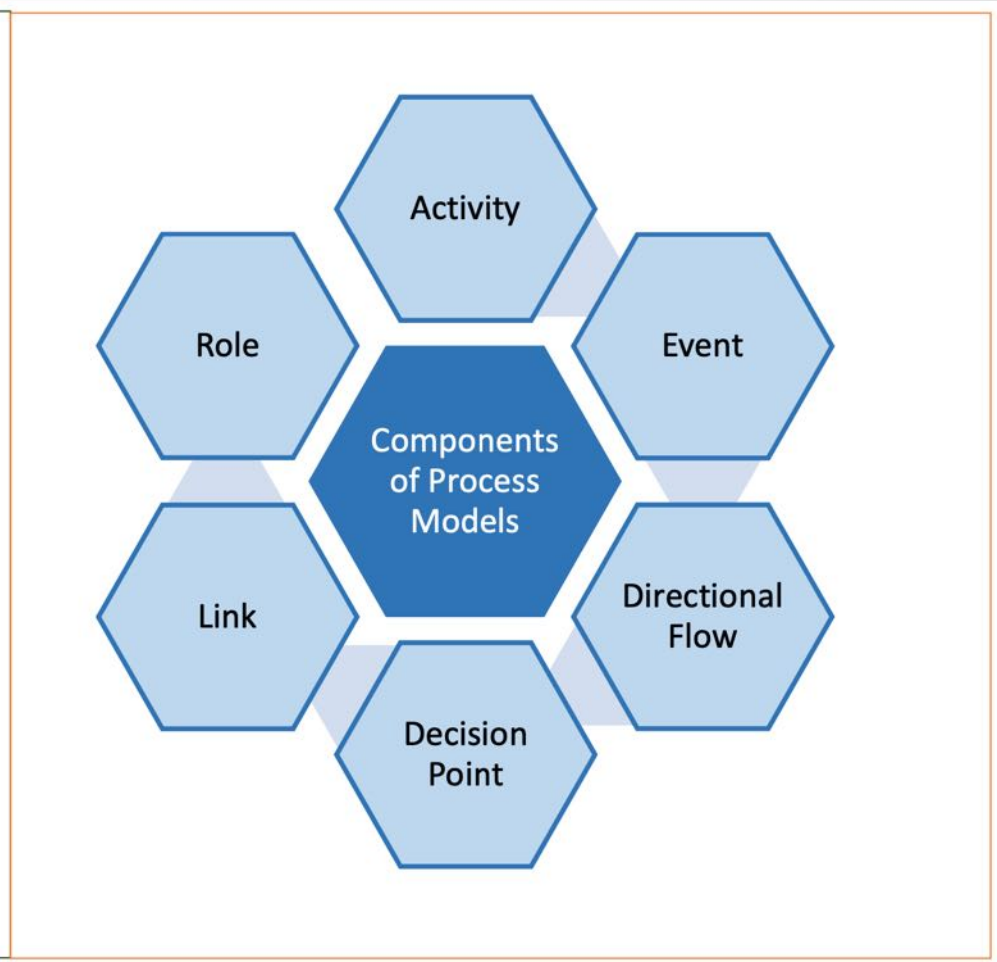
- Includes all entities and relationships between them
- Specifies attributes and a primary key for each entity
- Specifies foreign keys, which identify the relationship between different entities
- Involves normalization

PHYSICAL DATA MODELING

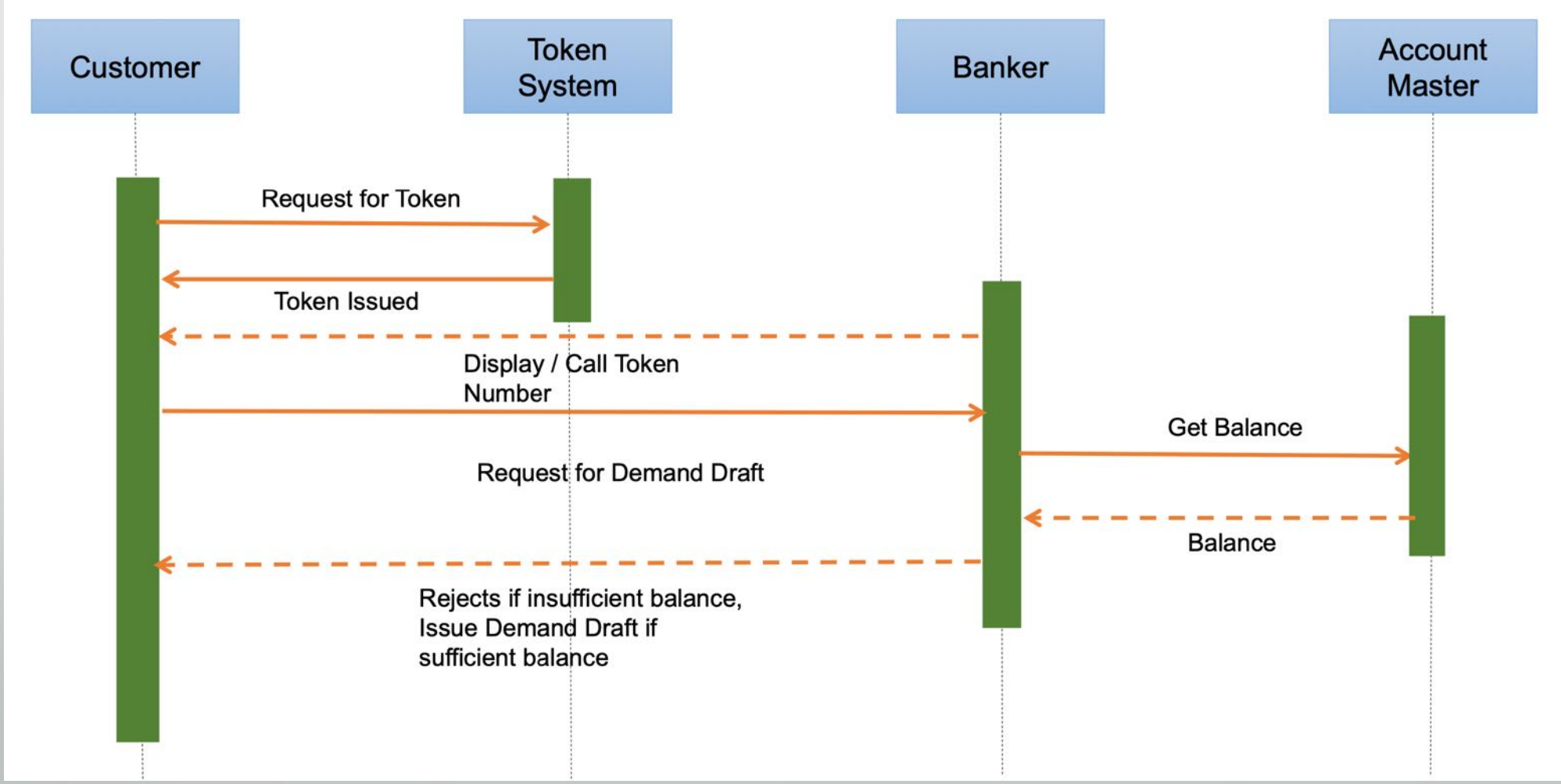
- Specifies all tables and columns
- Includes foreign keys to identify relationships between tables
- May include denormalization, depending on user requirements
- May be significantly different from the logical data model

PROCESS MODELING

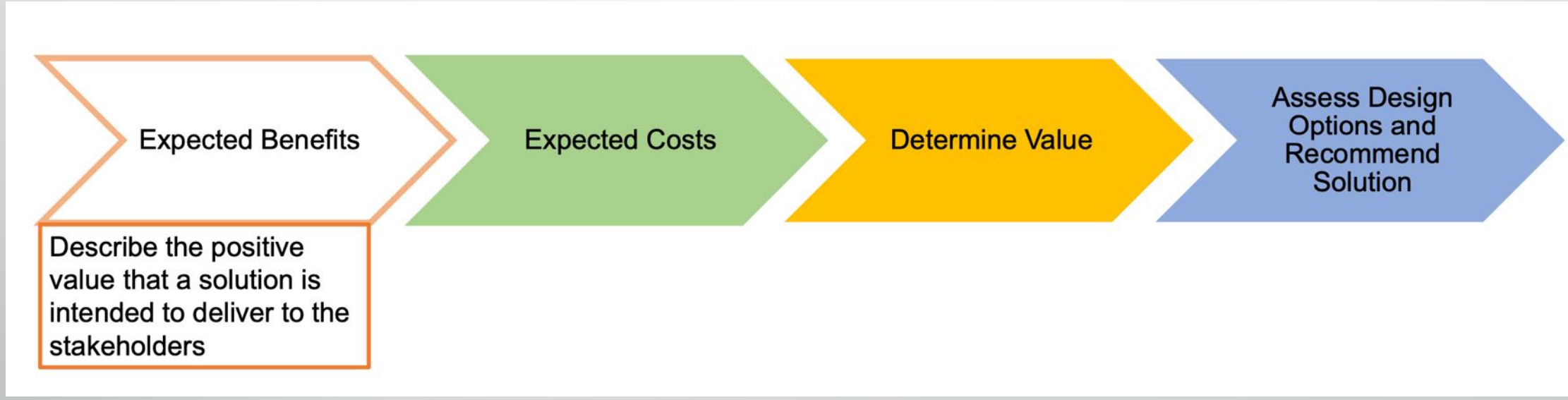




SEQUENCE DIAGRAM

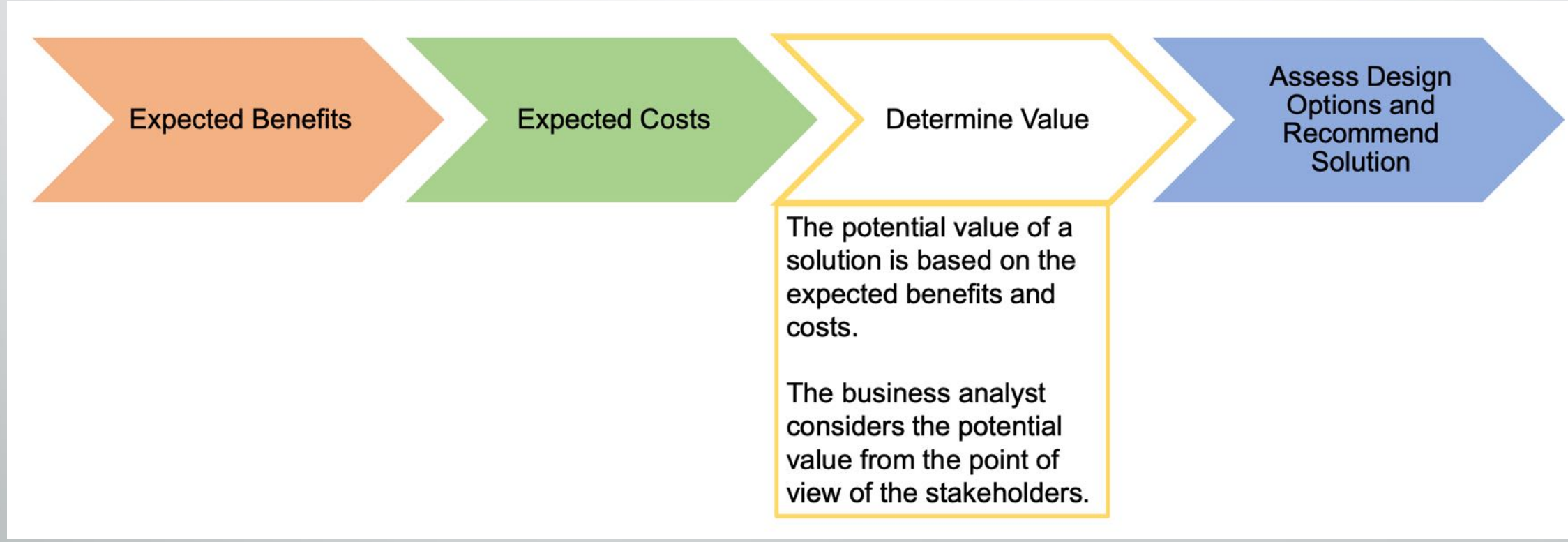


ANALYZE POTENTIAL VALUE

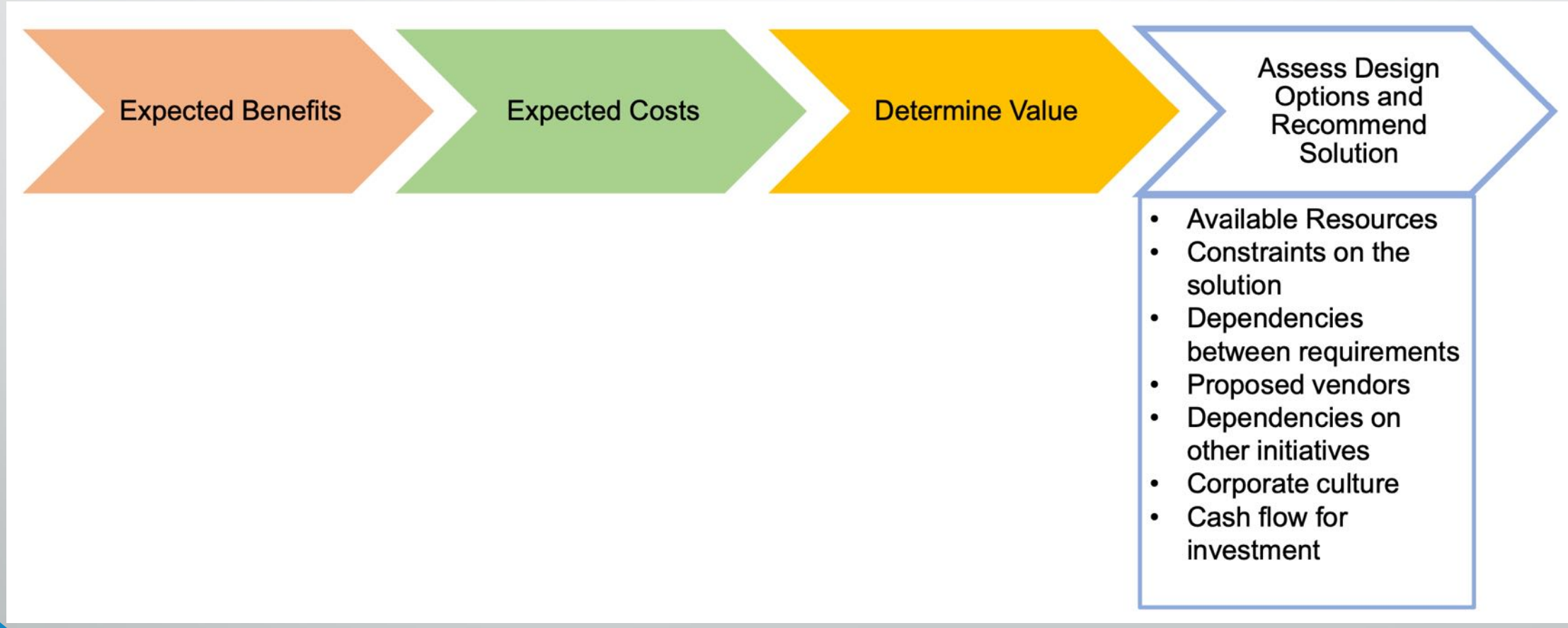


ANALYZE POTENTIAL VALUE

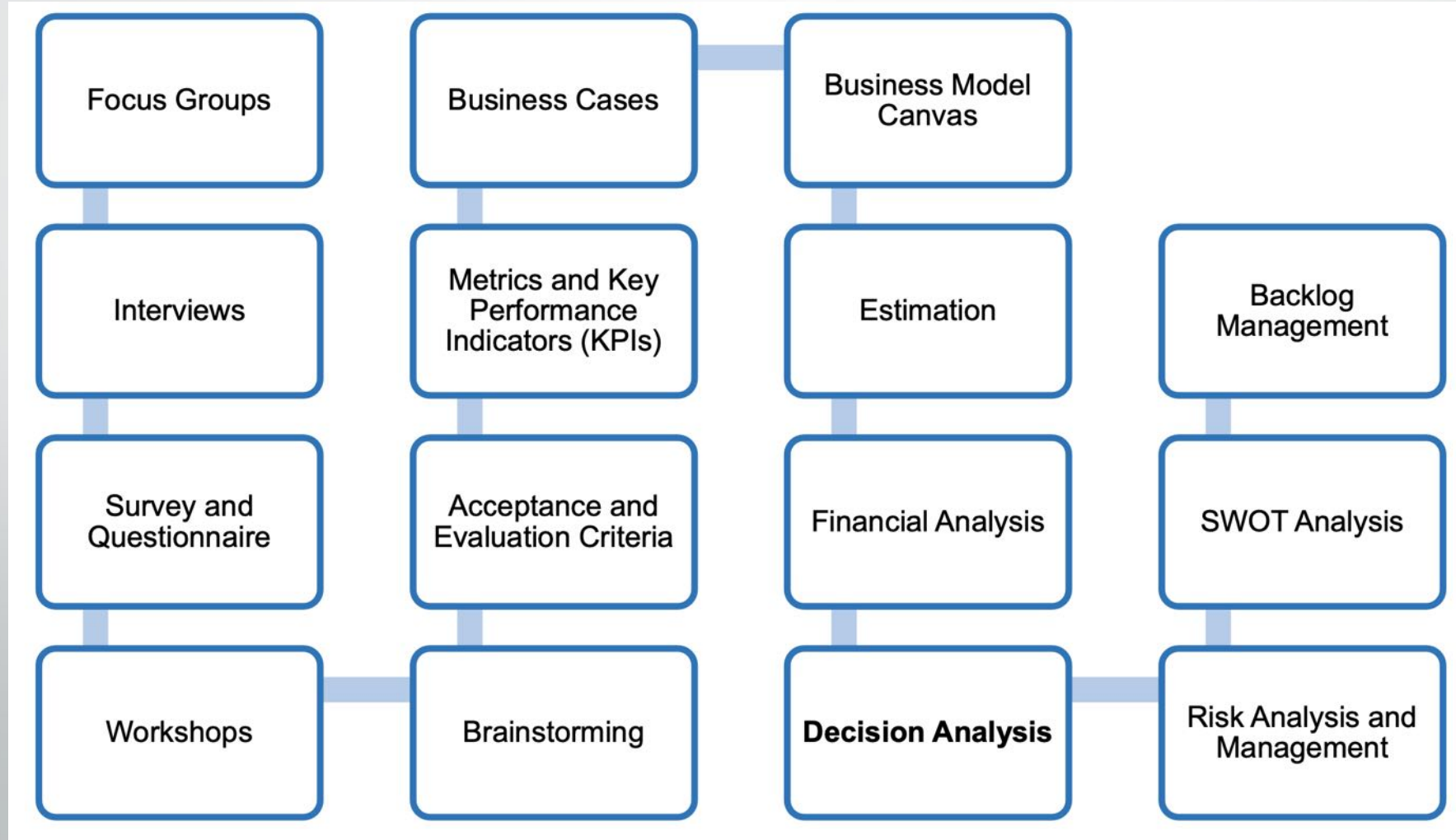
ANALYZE POTENTIAL VALUE



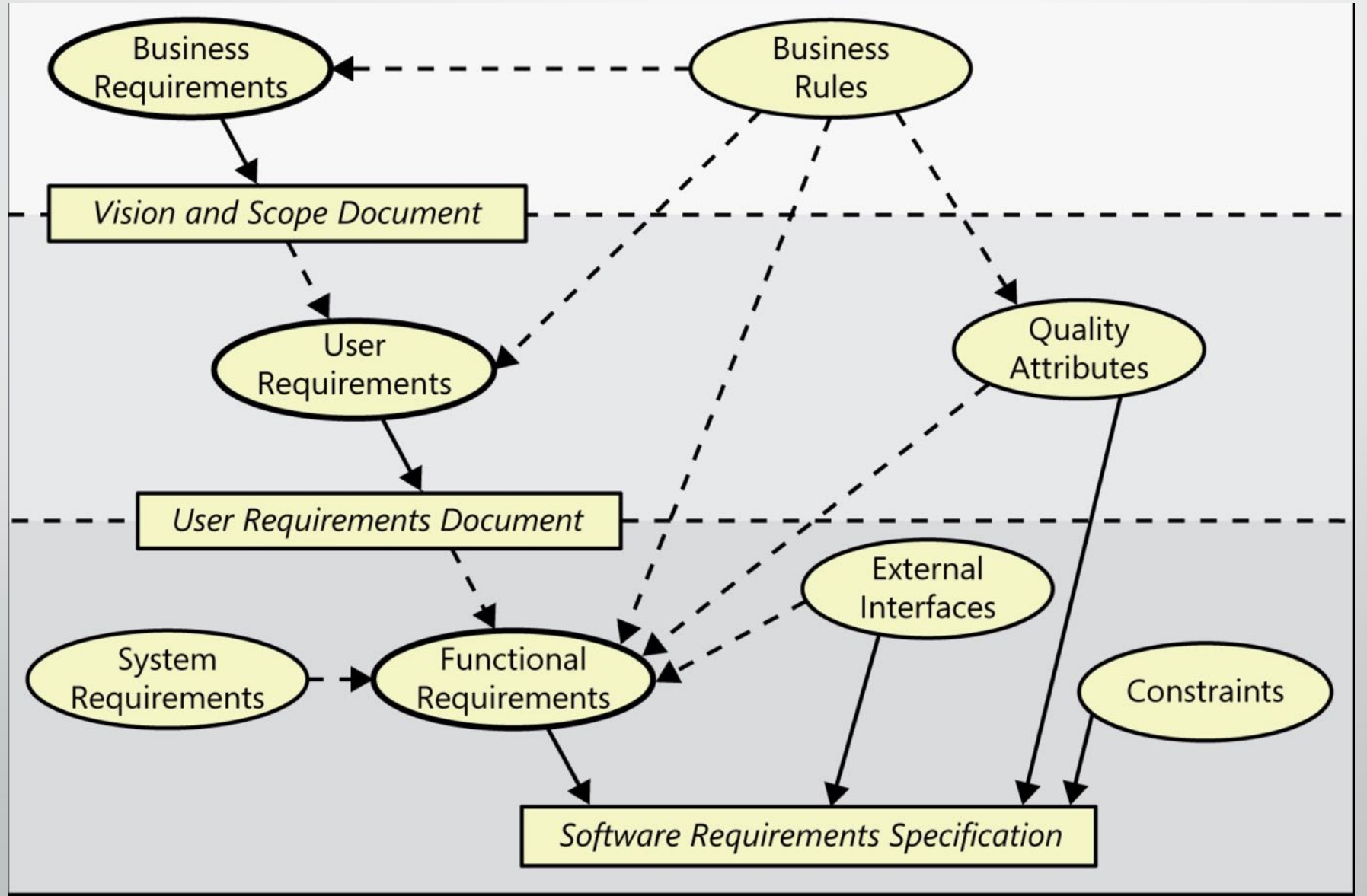
ANALYZE POTENTIAL VALUE



ANALYZE POTENTIAL VALUE - TECHNIQUES



Summary – sources of requirement



Summary – req. analysis

Who	• Who are the project's stakeholders?	• Stakeholder categories
	• Who will directly interact with the system?	• Actor table (and possibly an actor map) or personas
	• Who will see what when they interact with the system?	• Dialog map (supplemented with or substituted by a prototype or dialog hierarchy)
What	• What do important business terms mean?	• Glossary
	• What functions in the organization interact to share information?	• Relationship map (a business model)
	• What information or assets go into and out of the system?	• Context diagram
	• What are the static data elements that must be stored and how are they related?	• Data model (supplemented with or substituted by a class model, data tables, or a data dictionary)

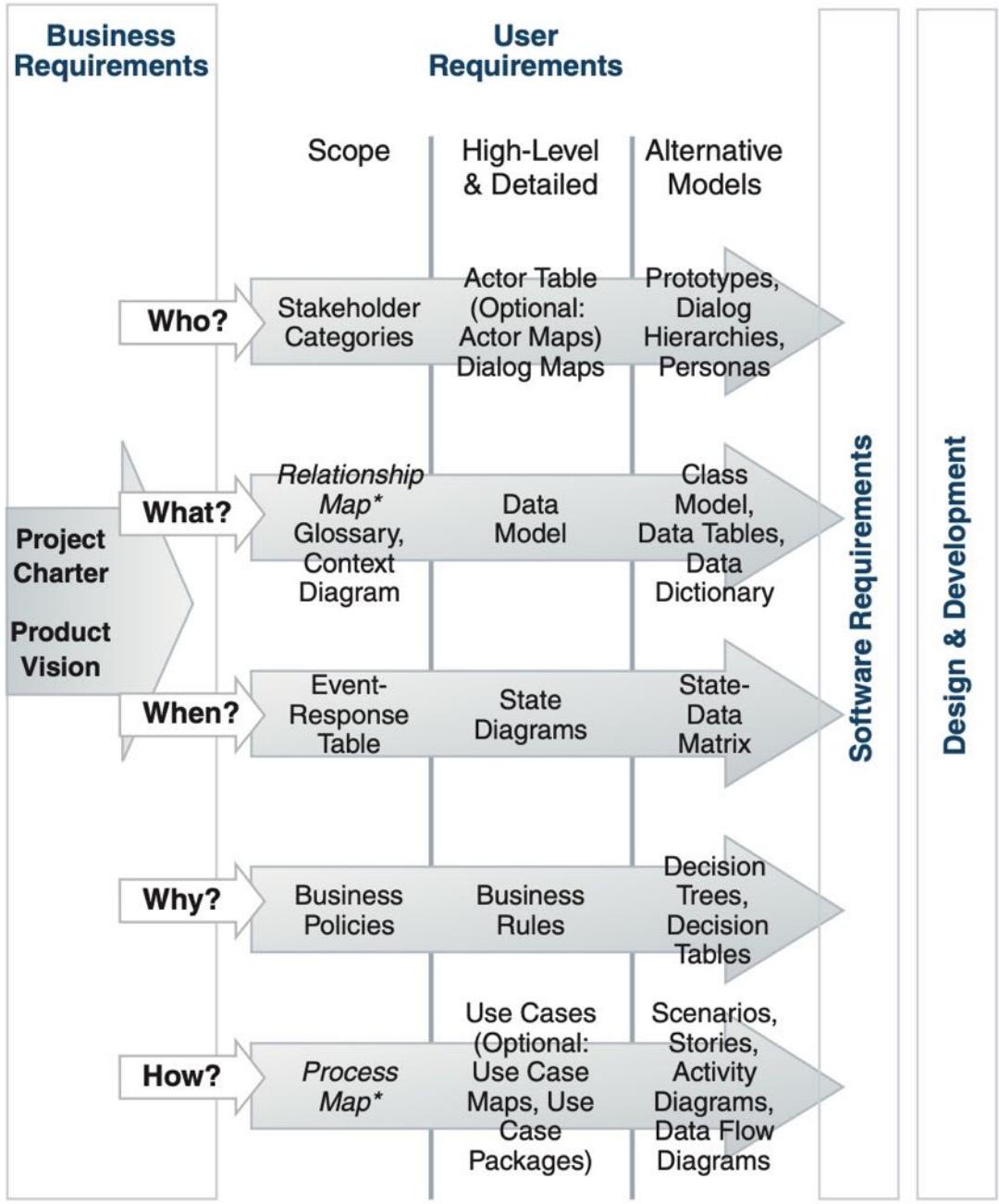
Summary – req. analysis

When	• When does the system need to respond or act? • When do tasks get performed and when does information change?	• Event-response table • State diagram (supplemented with or substituted by a state-data matrix)
Why	• Why are we motivated to enforce standards, policies, regulations, and legislation? • Why are the decisions made that influence behavior and assert business structure?	• Business policies • Business rules (supplemented with or substituted by decision tables or decision trees)

Summary – req. analysis

How	• How do processes operate in the business to achieve business goals? • How are tasks performed and in what sequence?	• Process map (a business model) • Use cases and possibly use case maps and use case packages (supplemented with or substituted by scenarios, stories, activity diagrams of use cases, or data flow diagrams)

Summary – User Requirements Models Roadmap



Adapted from Reference 4: Gottesdiener, 2002

Summary – Tools and techniques (for req. analysis)

When you need to:	Then create:
Model the business	Some combination of Relationship Map and/or Process Map
Understand the project scope	Some combination of Context Diagram, Event-Response Table, and/or Business Policies
Add detail to user requirements	Some combination or variation of Actor Table, Use Cases, Dialog Maps, Data Model, State Diagrams, and/or Business Rules
Negotiate trade-offs among requirements	Prioritized Requirements

Summary - Terms

1. **acceptance** criteria Conditions that a software product must satisfy to be accepted by a user, customer, or other stakeholder.
2. **assumption** A statement that is believed to be true in the absence of proof or definitive knowledge
3. **Backlog:** On an agile project, the prioritized list of work remaining for the project. A backlog can contain user stories, business processes, change requests, infrastructure development, and defect stories. Work items from the backlog are allocated to upcoming iterations based on their priority.
4. **baseline** A snapshot in time that represents the current agreed-upon, reviewed, and approved set of requirements, often defining the contents of a specific product release or development iteration. Serves as the basis for further development work
5. **business objective** A financial or nonfinancial business benefit that an organization expects to receive as a result of a project or some other initiative

Summary - Terms

6.**business rule** A policy, guideline, standard, regulation, or computational formula that defines or constrains some aspect of the business.
7. change control board (CCB) The group of people responsible for deciding to accept or reject proposed changes on a software project, including changes in requirements.
8. **class diagram** An analysis model that shows a set of system or problem domain classes, their interfaces, and their relationships.
9. constraint A restriction that is imposed on the choices available to the developer for the design and construction of a product. Other types of constraints can restrict the options available to project managers. Business rules often impose constraints on business operations and hence on software systems.
10.**external interface requirement** A description of a connection between a software system and a user, another software system, or a hardware device.

Summary - Terms

11. **COTS (commercial off-the-shelf)** A software package purchased from a vendor and either used as a self-contained solution to a problem or integrated, customized, and/or extended to satisfy customer needs.
12**. dashboard** report A screen display or printed report that uses multiple textual and/or graphical representations of data to provide a consolidated, multidimensional view of what is going on in an organization or a process.
13. **decision rule** An agreed-upon way by which a body of people arrives at a decision.
14. **event** A trigger or stimulus that takes place in a system's environment that leads to a system response, such as a functional behavior or a change in state.
15. **dependency** As used in requirements specification, a reliance that a project has on a factor, event, or group outside its control.

Summary - Terms

16. **external entity** An object in a context diagram or a data flow diagram that represents a user class, actor, software system, or hardware device that is external to the system being described but interfaces to it in some fashion. Also called a terminator.
17. assumption A statement that is believed to be true in the absence of proof or definitive knowledge.
18. **validation** The process of evaluating a project deliverable to determine whether it satisfies customer needs. Often stated as “Are we building the **right product?**”
And **verification** The process of evaluating a project deliverable to determine whether it satisfies
the specifications on which it was based. Often stated as “Are we building the **product right?**”

Summary - Terms

19. **prototype:** A partial, preliminary, or possible implementation of a software system. Used to explore and validate requirements and design approaches.
- **horizontal** prototype : mock-up: A partial or possible representation of a user interface for a software system. Used to evaluate usability and to assess the completeness and correctness of requirements. Could be executable or could be in the form of a paper prototype.
- **vertical** prototype: proof-of-concept: A prototype that implements a portion of a software-containing system that slices through multiple layers of the architecture. Used to evaluate technical feasibility and performance.
- evolutionary and throwaway
    + **evolutionary** prototype A fully functional prototype created as a skeleton or an initial increment of the final product, which is fleshed out and extended incrementally as requirements become clear and ready for implementation.
    + **throwaway**: A prototype that is created with the intent of discarding it after it has served its purpose of clarifying and validating requirements and/or design alternatives.
- paper and electronic
    + **paper:** A non-executable mock-up of a software system's user interface using low-tech screen sketches.
- **wireframe** A kind of throwaway mock-up prototype that is often used for preliminary webpage design

Summary - Terms

20. **gold-plating** Unnecessary or excessively complex functionality that is specified or built into a product, sometimes without customer approval.
21. **nonfunctional** requirement A description of a property or characteristic that a system must exhibit or a constraint that it must respect.
22. **product champion** A designated representative of a specific user class who supplies the user requirements for the group that he or she represents.
23. subject matter expert An individual who has extensive experience and knowledge in a domain and who is recognized as an authoritative source of information about the domain.
24.Scope creep: A condition in which the scope of a project continues to increase in an uncontrolled fashion throughout the development process.
25. vision and scope document A collection of the business requirements for a new system, including business objectives, success metrics, a product vision statement, and a project scope description.
vision A statement that describes the strategic concept or the ultimate purpose and form of a new system.

Summary - Terms

26. **risk** A condition that could cause some loss or otherwise threaten the success of a project.
27. **reuse**: The act of using existing requirements knowledge in multiple systems that share some similar functionality.
28. r**equirements traceability matrix**
A table that depicts logical links between individual functional requirements and other system artifacts, including other functional requirements, user requirements, business requirements, architecture and design elements, code modules, tests, and business rules.
29. **stakeholder** An individual, group, or organization that is actively involved in a project, is affected by its process or outcome, or can influence its process or outcome.
30. **retrospective** A review in which project participants reflect on the project's activities and outcomes with the intent of identifying ways to make the next project be even more successful.
31. **quality** attribute (**quality-of-service**) A nonfunctional requirement that describes a service or performance characteristic of a product. Types of quality attributes include usability, portability, maintainability, integrity, efficiency, reliability, and robustness. Quality attribute requirements describe the extent to which a software product must demonstrate desired characteristics.

Summary - Terms

32. **Analysis:** The process of classifying requirements information into various categories, evaluating requirements for desirable qualities, representing requirements in different forms, deriving detailed requirements from high-level requirements, negotiating priorities, and related activities.
- **activity** diagram An analysis model that depicts a process flow proceeding from one activity to another. Similar to a flowchart.
- **decision tree** An analysis model that visually depicts the actions a system takes in response to specific combinations of a set of conditions.
- **dialog map** An analysis model that depicts a user interface architecture, showing the dialog elements with which the user can interact and the navigations permitted between them.
- **entity-relationship diagram** An analysis model that identifies the logical relationships between pairs of entities. Used for modeling data
- **flowchart** An analysis model that shows the processing steps and decision points in the logic of a process. Similar to an activity diagram.
- **GAP analysis** A comparison of the current state to an alternative or potential state for a system, process, or other aspect of a business situation, to identify significant differences between them.
- **root cause analysis** An activity that seeks to understand the underlying factors that contribute to an observed problem

Summary - Terms

32. **Analysis (cont.)**
- context diagram An analysis model that depicts a system at a high level of abstraction. The context diagram identifies objects outside the system that exchange data with the system, but it shows nothing about the system's internal structure or behavior.
- data dictionary A collection of definitions for the data elements and data structures that are relevant to the problem domain.
- **data flow diagram** An analysis model that depicts the processes, data stores, external entities, and flows among them that characterize the behavior of data flowing through business processes or software systems.
- **state-transition diagram** An analysis model that shows the sequence of states that an object in a system goes through during its lifetime in response to specific events that take place, or that shows the possible states of the system as a whole.
- **swimlane** diagram An analysis model that shows the sequential steps of a business process flow or the operations of a proposed software system.

Summary - Terms

33. **pilot** A controlled execution of a new solution (such as a process, tool, software system, or training course) with the objective of evaluating the solution under real conditions to assess its readiness for general deployment.

34. **Use case detail**
alternative flow A path through a use case that leads to success but that involves a variation from the normal flow in the specifics of the task or in the actor's interaction with the system.
exception A condition that can prevent a use case from concluding successfully. Unless some recovery mechanism is possible, the use case's postconditions are not reached and the actor's goal is not achieved.
postcondition A condition that describes the state of a system after a use case is successfully completed.
precondition A condition that must be satisfied or a state the system must be in before a use case can begin.
extend relationship A construct in which an alternative flow in a use case branches off from the normal flow into a separate extension use case.

Summary - Story

"On Monday, Oct 9, Mr. Long leaves a message that he wants to reschedule the cleaning scheduled for Tuesday, Oct 10, and he wants to add two inside mirror cleanings to the job.

He requests a phone estimate and a repeat customer discount.

We told him it would cost $250 after the 10% discount.

We give him the next available day (the following Fri at 3:00 p.m.) with the same cleaner (Ms. Hong) that he had six months ago.

He asked for an earlier date, and we gave him Wednesday with another contractor (Mr. Minh) at 10 a.m.

We confirm that he'll pay by credit card at the time of service, and we read back his credit card number to him on the phone."

Summary - Diagram

PFPT Software is a large outsourcing company in Vietnam. They develop various kinds of software for various customers. After the product goes live, a customer support ticket system will handle customer feedback. When a customer raises a ticket, the supporter will assign it to the relevant team to analyze and develop or fix it. The solution must be tested by the QA team before integrating it into the software. Each team has many team members, and they work 24/7 to give the best support to customers. Of course, they have to split shifts to do this. Then, when everything is done, the development team will resolve the ticket and the customer will be notified about that.

- Q1. List >= 3 actors
- Q2. List >= 7 use cases' names
- Q3. Give the details of "deliver resolved ticket" use case
- Q4. Give swimlane diagram for the above scenario
- Q5. Give >=4 state of ticket
- Q6. Give state machine diagram for ticket

Summary - Diagram

ADS is a system using by real estate company. By gathering all pertinent and brief details about real estate that are connected to real-world photos, the real estate post-managing system satisfies the client's needs regarding renting and purchasing real estate for themselves. Customers can save time by going straight to the real estate area with the system's assistance and can more quickly compare pricing amongst reality options. The broker has more internet access to potential consumers who can be profitable. Customers can simply pay fees through the payment gateway; At the same time, it will be completely easy for brokers to create ads to reach people to rent or buy. Through linking with advertising platforms such as Facebook and Google, any advertisement can be included in campaigns to run ads. Which campaigns to run are configured by the broker on the system, and payments are also easily managed by configuring the maximum amount and number of days to run ads.

- Q1. List >= 3 actors
- Q2. List >= 7 use cases' names
- Q3. Give the details of "Add post to campaign" use case
- Q4. Give context diagram for the above scenario
- Q5. Give states of post
- Q6. Give state machine diagram for a post

Summary – Sample for PE

When developing a new software, the software development team must describe the software requirements in software requirements specification before they do designing and coding. They describe the software requirements by using text, formulas, drawing models and table description. The software tools department at FPT University want to develop a new software tool named FU Library Management System (FULIBOL) website in order to facilitate FPT University side and students side intercommunication. The FULIBOL website need to have many functions that allow importing data on student list from fap.fpt.edu.vn website, signing in FULIBOL by using @fpt.edu.vn email accounts of Students or Teachers or Librarian. The FULIBOL can exchange data through the library.books24x7.com website. Students can see the News on updated books added to the FULIBOL system and they can search the books they want to borrow. Students can receive the notification email reminding them the deadline of book borrowing 3 days before the deadline. Students who already logged in to the FULIBOL system can see the list of the books they borrow and the deadline of returning the books. They can request the Librarian to reset the new deadline of the books if they want to extend the borrowing time by sending email to Librarian. Besides, FULIBOL can record history of the books which students already borrowed. If the books have deadline over and the students do not request the new deadline or return the books to the Librarian, the students will receive automatic notification through email about the penalty money for the books they borrowed. Librarians can add one new book through the add book screen or import from the excel file new books list into the FULIBOL system. Staff can also add or import new books. Use case approach is one of the most effective ways to describe the software requirements.

Summary – Sample for PE (cont.)

Question 1: Complete all parts of the first page of the software requirements specification template provided you. (1 point)

Question 2: (4 points) Use case diagram is an effective way to visualize the interaction between actors and the software system. One Use case diagram is described by one rectangle which represents one system or one software application. One oval represents one use case. The name of the use case must begin with the verb and be followed by an object.

Draw 1 use case diagram (by using Visual Paradigm offline, students must insert the use case diagram based on the content of this exam paper in an image file in to provided template file) and the image must include the following information: **[NOTE. Student should not keep the flowing in the final answer of your exam]**

1. In this exam paper, the name of the rectangle is: ……(0.1 point)
2. The actor may be human, other software systems or devices. In this exam paper, list the name of >= 4 actors is: ……..(0.6 point)
3. In this exam paper, list the name of >= 5 use cases are: …….(1 point)
4. The actors are inside or outside of the rectangle? …… (0.1 point)
5. The ovals represent the use cases are inside or outside of the rectangle? ……(0.1 point)
6. The primary actor of the use case **Add Book** is: …….(0.2 point)

Summary – Sample for PE (cont.)

7. The secondary actor of the use case **Add Book** is:..... (0.2 point)
8. Based on the fact you already used http://ds.libol.fpt.edu.vn website. The use cases in this exam paper may have relationship.
   8.a List the name of two use cases that have **extend** relationship: .....(0.3 point)
   8.b The name of **base use case (in 8.a)** is: .....(0.2 point)
   8.c The name of **extend use case (in 8.a)** is: .....(0.2 point)
   8.d The dashed line with an arrow that points to which use case? .....(student write the name of use case) (0.1 point)
9. Based on the fact you already used http://ds.libol.fpt.edu.vn website, the use cases in this exam paper may have relationship.
   9.a List the name of two use cases that have **include** relationship: .....(0.3 point)
   9.b The name of **based use case (in 9.a)** is: .....(0.2 point)
   9.c The name of **included use case (in 9.a)** is: .....(0.2 point)
   9.d The dashed line with an arrow that points to which use case? ..... (student write the name of use case) (0.2 point)

Summary – Sample for PE (cont.)

Question 3: Writing detail 1 use case **Add Book** in template provided with this exam paper. (4 points)

Question 4: writing 2 non-functional requirements. (1 point)

a. The term of the most important non-functional requirements and at least 2 statements for the FULIBOL system. Should be specific in number, explain the number so that the tester can test the software and make conclusion the test is passed or failed. (0.5 point). **The answers to this question will get ZERO if your answers have any keywords not related to this exam paper.**

b. The term of the second important non-functional requirement and at least 2 statements for the FULIBOL system. Should be specific in number, explain the number so that the tester can test the software and make conclusion the test is passed or failed. (0.5 point). **The answers to this question will get ZERO if your answers have any keywords not related to this exam paper.**

Summary – Feature tree

- **feature**: One or more logically related system capabilities that provide value to a user and are described by a set of functional requirements.
- **feature tree**: An analysis model that depicts the features planned for a product in a hierarchical tree, showing up to two levels of sub-features beneath each main feature.

Content portal has below feature
- Registration
- Account management
- Content Access
- Content management

Analyze this using feature tree

Summary – Practice

The company NewFutureVBook wants to build a web-based book rental business whereby customers interact with the company using the Internet. Customers use a web page to browse the book inventory, check the book's availability, and request to borrow book (up to three books at a time). The company mails requested books out to customers, and customers mail the books back when they have finished reading them. The rental period is three weeks. Books have different rental fees based on their release date, or type, etc. If a book is returned after the loan’s due date, the borrower will be charged a fine of 1000 VND for each late day.

Heavily-used books may be put on reserve, meaning that members can read them only in the library and cannot borrow them.

When an account's balance falls below the value Minimum Balance, the account will be suspended and the customer cannot initiate new loans until payment is made and the balance becomes nonnegative. Information about both current and past loans is maintained.

1. Provide a context diagram for the system
2. Provide a use-case diagram that depicts only those use cases that are initiated by company staff
3. Draw a class diagram for the above problem