CHAPTER -7

Requirements Engineering Tools

Compiled By:Shambel Adugna

Requirements Engineering Tools

- The creation of a natural-language software requirements specification is helpful to contain the **functional** and **nonfunctional** requirements, as well as documents that contain the business requirements and user requirements.
- .We pointed out that these deliverables are just containers for sets of requirements information; they need not be traditional word-processing documents.

Requirements Engineering Tools…

- **Although still widely used, a document-based approach to developing and managing**

requirements has numerous limitations, including the following:-

- It’s difficult to keep the documents current and synchronized.
- Communicating changes to all affected team members is a manual process.
- It’s not easy to store supplementary information—attributes—about each requirement.
- It’s hard to define links between requirements and other system elements.
- Tracking the status of both individual requirements and the entire set of requirements is

cumbersome.

Requirements Engineering Tools…

■Reusing a requirement generally means that the business analyst must copy the text from

the original document into another document for each other system or product where the

requirement is to be used.

■It’s difficult for multiple project participants to modify the requirements, particularly if the

participants are geographically separated.

■There’s no convenient place to store proposed requirements that were considered but rejected

■ Identifying missing, duplicate, and unnecessary requirements is difficult.

Requirements Engineering Tools…

- **Requirements development (RD) tools** and **requirements management (RM) tools** provide **solutions** to all of these **limitations**.
- **RD tools** can help you elicit the right requirements for your project and judge whether those requirements are well-written.
- **RM tools** help you manage changes to those requirements, track status, and trace requirements to other project deliverables.

Requirements Engineering Tools…

1. Requirements Development (RD) Tools

- **Requirements development (RD) tools**: are used by business analysts to work with stakeholders to elicit and document requirements more effectively and more efficiently than with manual methods.
- Stakeholders will vary in how they best consume and share information: **textually**, **visually**, or **audibly**.

Requirements Engineering Tools…

1. Requirements Development (RD) Tools…

- **RD tools** can improve stakeholder collaboration by accommodating a variety of communication methods.
- We can divides the development tools into **elicitation**, **prototyping**, and **modeling** tools.
- Some of the tools in the RD category provide all of these services.
- Some of them also offer requirements management capabilities.
- In general, RD tools are not as mature as RM tools, and their overall impact on projects is typically less than that of RM tools.

1. Requirements Development (RD) Tools…

1. Elicitation tools

- Elicitation tools include those used for recording notes during elicitation sessions.
- These enable the BA to quickly organize ideas and to annotate follow-up questions, action items, core terms, and the like.
- Mind-mapping tools facilitate brainstorming as well as organizing the information produced.

2. Prototyping tools

- Prototyping tools facilitate the creation of work products that range from electronic mock-ups to full application simulations.
- Common applications such as Microsoft PowerPoint can be used to quickly mock up screens and the navigations between them or to annotate existing screen shots.

1. Requirements Development (RD) Tools…

1. Modeling tools

- These tools support the use of standard **shapes**, **notations**, and **syntax** for drawing diagrams according to established conventions.
- They also enable you to create diagrams that look cleaner and more consistent than if you draw them manually.
- Specialized software modeling tools facilitate iteration by dragging along connected arrows and labels whenever you move a symbol in the diagram; general-purpose drawing tools might not provide that capability.

2. Requirements management tools

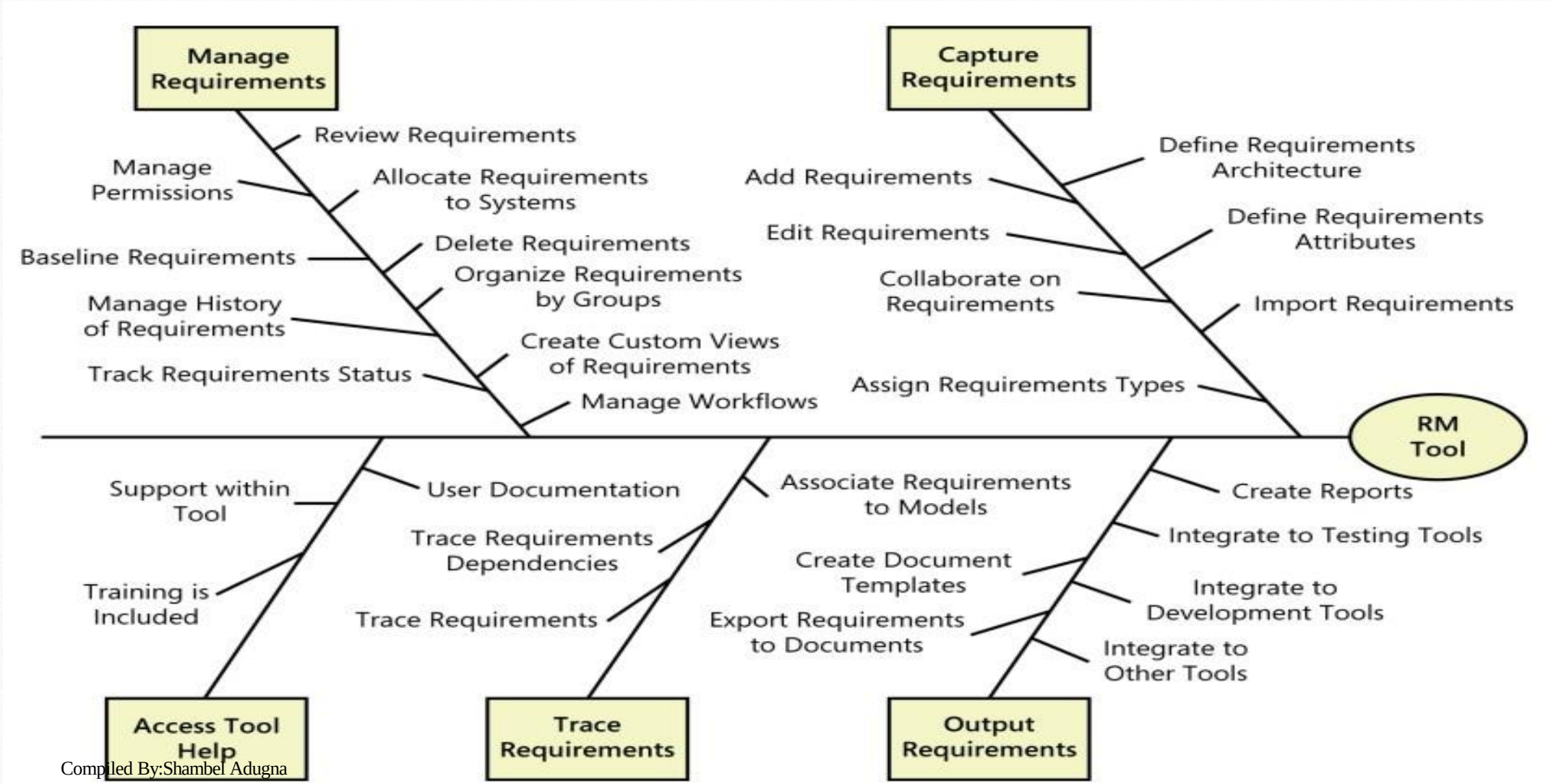
- An **RM tool** that stores information in a multiuser database provides a robust solution to the limitations of storing requirements in documents.
- Small project teams can get away with just entering the requirements text and several attributes of each requirement.
- Larger project teams will benefit from letting users import requirements from source documents, define attribute values, filter and display the database contents, export requirements in various formats, define traceability links, and connect requirements to items stored in other software development tools.

2. Requirements management tools …

❖ **Benefits of using an RM tool**

- ✓ Manage versions and changes
- ✓ Store requirements attributes
- ✓ Facilitate impact analysis
- ✓ Identify missing and extraneous requirements
- ✓ Track requirements status
- ✓ Control access

2. Requirements management tools(RM tool capabilities)…



2. Requirements management tools…

- **RM tools** let you define different requirement types, such as business requirements, **use cases**, **functional requirements**, hardware requirements, and **constraints**.
- This lets you differentiate all the types of information that are typically contained in an SRS.
- Many tools allow you to configure an information architecture (which defines how requirements types and other objects relate to one another) that is customized to your practices.

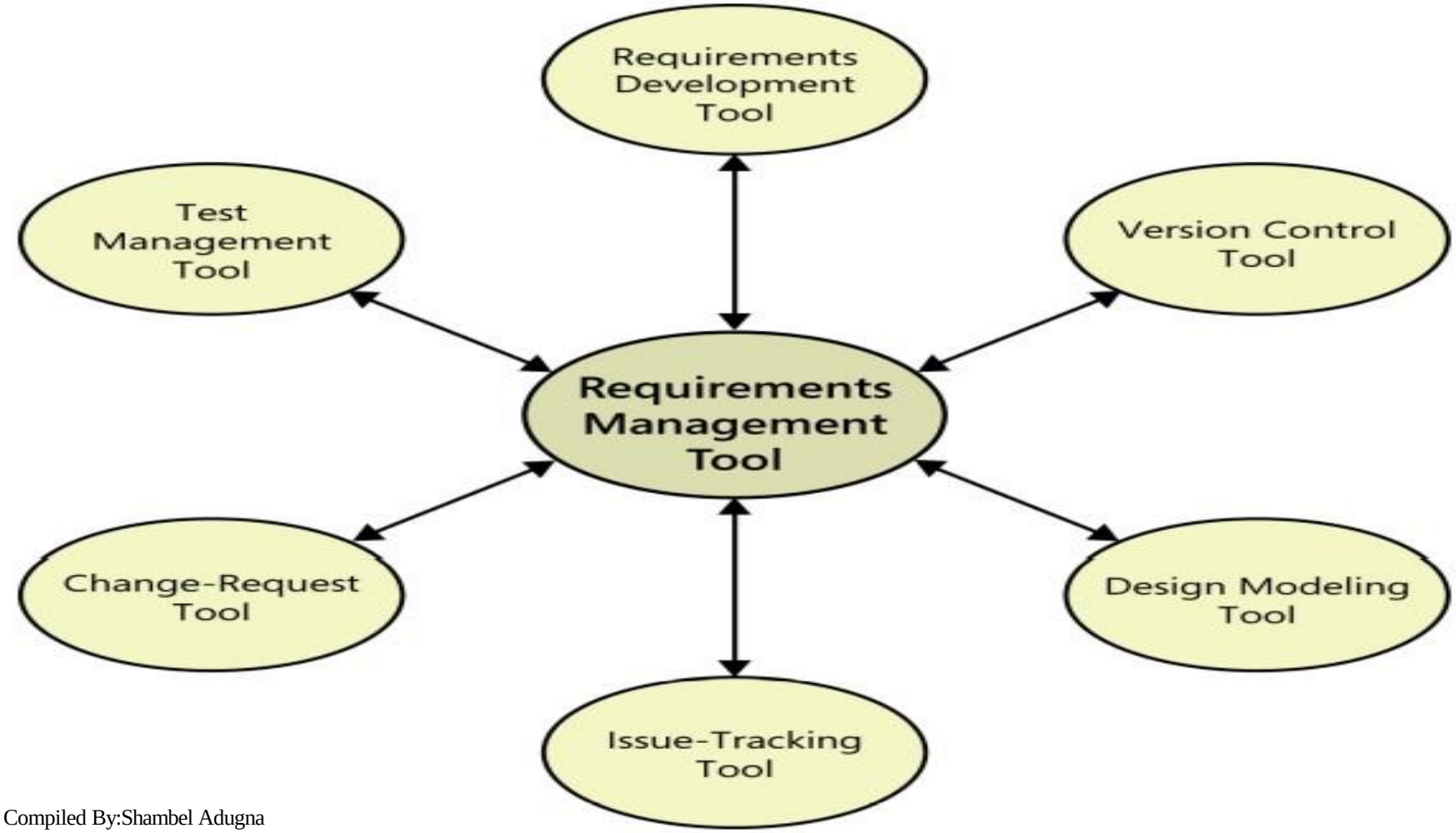
2. Requirements management tools…

- **Requirements management tools** generally have robust tracing features.
- **Tracing** is handled by defining links between two types of objects or objects of the same type.

- Some requirements management tools include modeling capabilities that also allow the models to be linked at an element level to individual requirements or to other model elements.
- Some agile project management tools also provide RM capabilities.
- These tools are used to manage and prioritize backlogs, allocate requirements to iterations, and generate test cases directly from requirements.

2. Requirements management tools…

- RM tools often integrate with other tools used in application development.



2. Requirements management tools…

- **Selecting and implementing a requirements tool**
- Any of these requirements tools can move your requirements practices to a higher plane of sophistication and capability.
- However, success depends upon selecting the most appropriate tool for your organization and getting your teams to adopt it as part of their routine practices.

1. **Selecting a tool:-** Select a tool based on the combination of desired features, platform, and pricing that best fits your development environment and culture.

✓ Business analysts should lead the selection efforts by defining the evaluation criteria and performing the actual assessment.

2. Requirements management tools…

- To summarize the selection process:-

Identify your organization's requirements for the tool to serve as evaluation criteria.

2.Prioritize and weight the criteria according to what capabilities or other factors matter most to your organization.

3. Set up demos or acquire evaluation copies of the tools you want to consider.

4. Score each tool against the criteria in a consistent manner.

5. Calculate a total score for each tool by using your criteria scores and the weights you assigned to them.

PART - II

Compiled By:Shambel Adugna

Requirements Engineering Techniques

- **Methods for RE**
- **Viewpoint-oriented Requirements Methods**

Compiled By:Shambel Adugna

Requirements Engineering Techniques…

- Requirements Engineering Methods (REMs) **support Requirements Engineering (RE) tasks, from elicitation, through modeling and analysis, to validation and evolution of requirements**.
- Despite the growing interest to design, validate and teach REMs, it remains unclear what components REMs should have.

Requirements Engineering Techniques…

- No single method or technique will be sufficient.
- Three common types of failure are examined, **process failure**, **expectation failure** and **interaction failure.**

Requirements Engineering Techniques…

- **Viewpoints** focus primarily on the early elicitation stages of the requirements process.
- Systematic support has always been harder to provide here than for system modelling or specification.
- using viewpoints can help structure the elicitation process.
- Viewpoints can also help prioritise and manage requirements.

Requirements Engineering Techniques…

- Another way of thinking of viewpoints is that they: -

  - ✓ focus the analyst's attention on the parts of the problem which affect stakeholders, and project the discovered requirements onto the system to be developed.

  - **Viewpoints on the problem and the system can be shown on the following figure:-**

Requirements Engineering Techniques…

- Stakeholders represent different ways of looking at a problem or problem **viewpoints**.
- This **multi-perspective analysis** is important as there is no single correct way to analyse system requirements.

Types of viewpoint

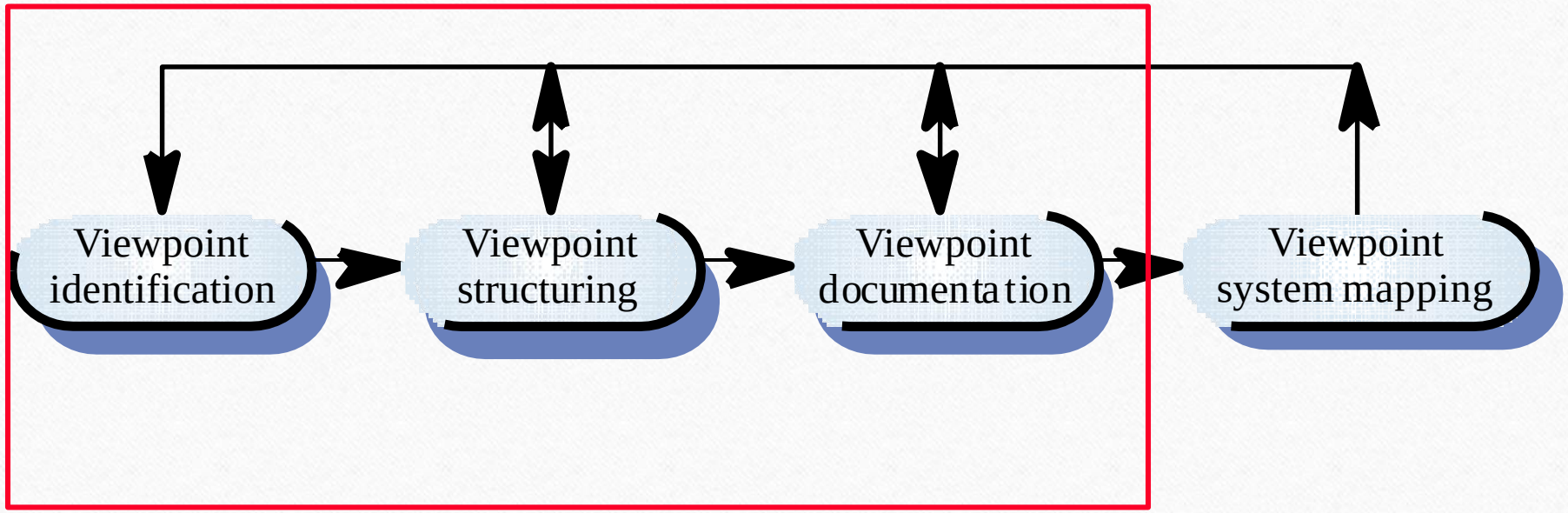
- **Data sources or sinks**
  - Viewpoints are responsible for *producing or consuming data*. Analysis involves checking that the data produced is actually consumed and viceversa
- **Representation frameworks**
- Viewpoints represent particular *types of system model*. These may be compared to discover requirements that would be missed using a single representation.
- **Receivers of services**
- Viewpoints are *external* to the system and *receive services* from it. Most suited to *interactive systems*.

External viewpoints (receivers of services)

- **Natural way to structure requirements elicitation**
  - ✓ they directly correspond to stakeholders and end-users
- **It is relatively easy to decide if a viewpoint is valid:**
  - ✓ it must interact with the system
- **Viewpoints and services provide a framework for structuring non-functional requirements**
  - ✓ non functional requirements may be associated to a service of a viewpoint, and
  - ✓ different viewpoints may have same service but different associated non functional requirements

The **VORD** method

- **V**iewpoint **O**riented **R**equirements **D**efinition



VORD process model

- **Viewpoint identification**
  - Discover **viewpoints** which receive system **services** and identify the services provided to each viewpoint
- **Viewpoint structuring**
  - Group related viewpoints into a **hierarchy**. Common services are provided at higher-levels in the hierarchy, and inherited by lower levels
- **Viewpoint documentation**
  - Refine the description of the identified viewpoints and services
- **Viewpoint-system mapping**
  - Transform the analysis to an **object-oriented design.**

VORD standard forms

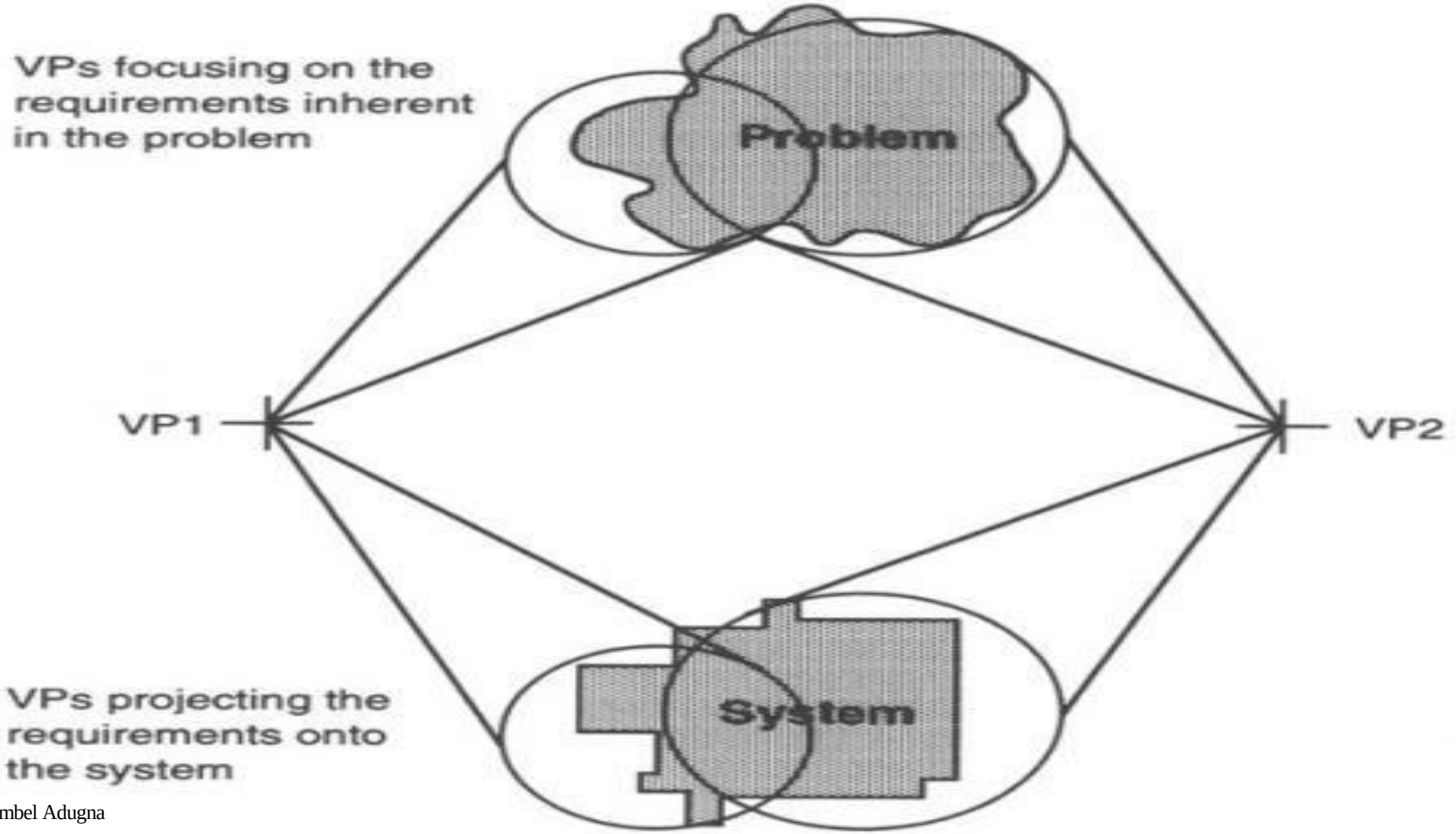
Viewpoint template

Reference:	The viewpoint name.
Attributes:	Attributes providing viewpoint information.
Events:	A reference to a set of event scenarios describing how the system reacts to viewpoint events.
Services	A reference to a set of service descriptions.
Sub-VPs:	The names of sub-viewpoints.

Service template

Reference:	The service name.
Rationale:	Reason why the service is provided.
Specification:	Reference to a list of service specifications. These may be expressed in different notations.
Viewpoints:	List of viewpoint names receiving the service.
Non-functional requirements:	Reference to a set of non-functional requirements which constrain the service.
Provider:	Reference to a list of system objects which provide the service.

Requirements Engineering Techniques…



Requirements Engineering Techniques…

- **Why Viewpoints?**
- A **viewpoint-based** approach to requirements analysis recognizes that all information about the system requirements cannot be discovered by considering the system from a single perspective.
- Any single perspective inevitably emphasises the requirements of one stakeholder at the expense of other viewpoints.
- **Without** a good definition of viewpoint, viewpoint identification is hard.
- There are also problems in integrating viewpoints' requirements and handling inconsistency in large projects.

PART - III

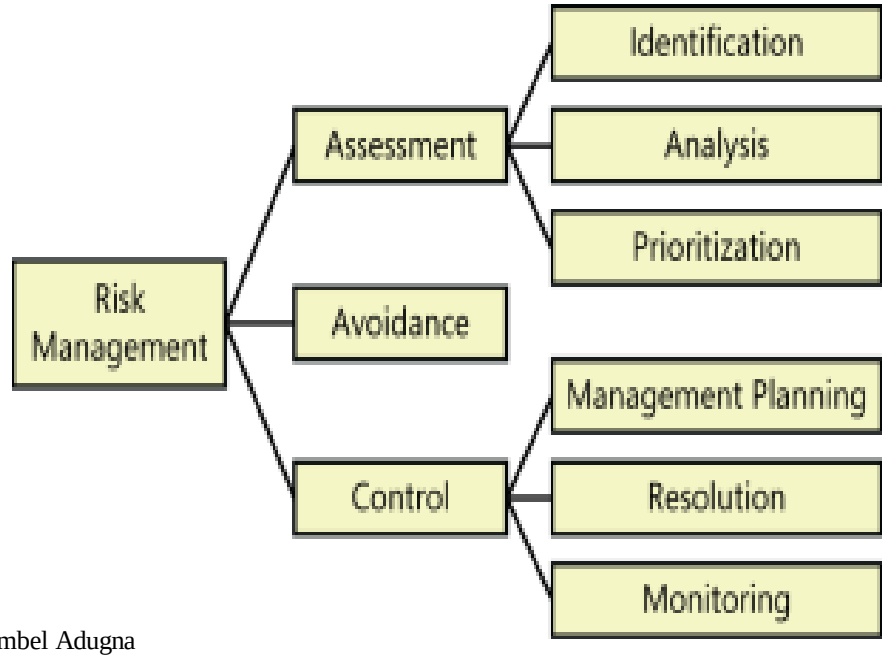
Compiled By:Shambel Adugna

Software Requirements and Risk Management

- **Projects** face many kinds of risks besides those related to requirements.
- Dependence on an external entity, such as a subcontractor or another project that is providing components to be reused, is a common source of risk.
- Project management is fraught with risks from **poor estimation**, **rejection of accurate estimates by managers**, **insufficient visibility into project status**, and **staff turnover.**
- Technology risks threaten highly complex and leading-edge development projects.

Software Requirements and Risk Management..

- Risk management involves the application of tools and procedures to contain project risk within acceptable limits.
- It provides a standard approach to identify and document risk factors, evaluate their potential severity, and propose strategies for mitigating them.



Elements of risk management.

Software Requirements and Risk Management..

- ***Risk assessment:*** is the process of examining a project to identify potential threats.
- During risk analysis, you'll examine the potential consequences of specific risks to your project.
- Risk prioritization helps you focus on the most severe risks by assessing the potential risk exposure from each.
- ***Risk avoidance:*** is one way to deal with a risk: don't do the risky thing.
- You can avoid some risks by not undertaking certain projects, by relying on proven rather than cutting-edge technologies and development methods, or by excluding features that will be especially difficult to implement.
- Most of the time you'll have to perform ***risk control*** activities to manage the top-priority risks you identified.

Documenting project risks

- It’s not enough to simply recognize the risks that face your project. You need to manage them in a way that lets you communicate risk issues and status to stakeholders throughout the project’s duration.

ID:
<sequence number>

Submitter:
<individual who brought this risk to the team's attention>

Date Opened:
<date the risk was identified>

Date Closed:
<date the risk was closed out>

Risk Statement:
<statement of the risk in the form "condition-consequence">

Scope of Impact:
<project teams, business areas, and functional areas the risk could affect>

Probability:
<the likelihood of the risk becoming a problem>

Impact:
<numerical rating of the potential damage if the risk does become a problem>

Exposure:
<probability multiplied by impact>

Risk Management Plan:
<one or more approaches to control, avoid, minimize, or otherwise mitigate the risk>

Contingency Plan:
<course of action to follow if the risk management plan is not effective>

Owner:
<individual responsible for resolving the risk>

Date Due:
<date by which the mitigation actions are to be implemented>

END !!

Compiled By:Shambel Adugna