

# **Course: Software Requirements Engineering**

## **Week 13: Tools & Agile Requirements Engineering**

**Lecturer: Yimer Amedie (MSc.)**

Addis Ababa Science and Technology University

May, 2026

# Contents



- Introduction
- Requirement Engineering Tools
- Requirement Development (RD) Tools
- Requirement Management (RM) Tools
- Agile Requirements Engineering Overview

**Figure 1.** *The software requirements*

**Note.** Image generated using Sora by OpenAI (2026).

# Learning Outcomes

After completing this lesson, you will be able to:

- Explain the role of Requirements Engineering tools.
- Use Requirements Development tools effectively.
- Apply Requirements Management tools for requirement control.
- Utilize traceability tools to track requirement relationships.
- Apply Agile Requirements Engineering tools and practices

# Introduction

Requirements documents alone become difficult to manage (Beatty, 2013)

✓ **Common challenges:**

- ✓ Requirement renumbering issues
- ✓ Requirement deletion and reuse
- ✓ Tracking ownership and status
- ✓ Managing multiple releases
- ✓ Maintaining traceability

# Introduction ... Cont'd

## Problems with Document-Based Requirements



1 Difficult to keep documents updated and synchronized



2 Changes must be communicated manually



3 Hard to store requirement attributes



4 Limited traceability between requirements and system elements



5 Tracking requirement status is cumbersome

# Introduction... Cont'd

## Additional Limitations



Managing requirements across multiple releases is difficult



Reusing requirements requires copying and pasting



Collaboration is challenging, especially for distributed teams



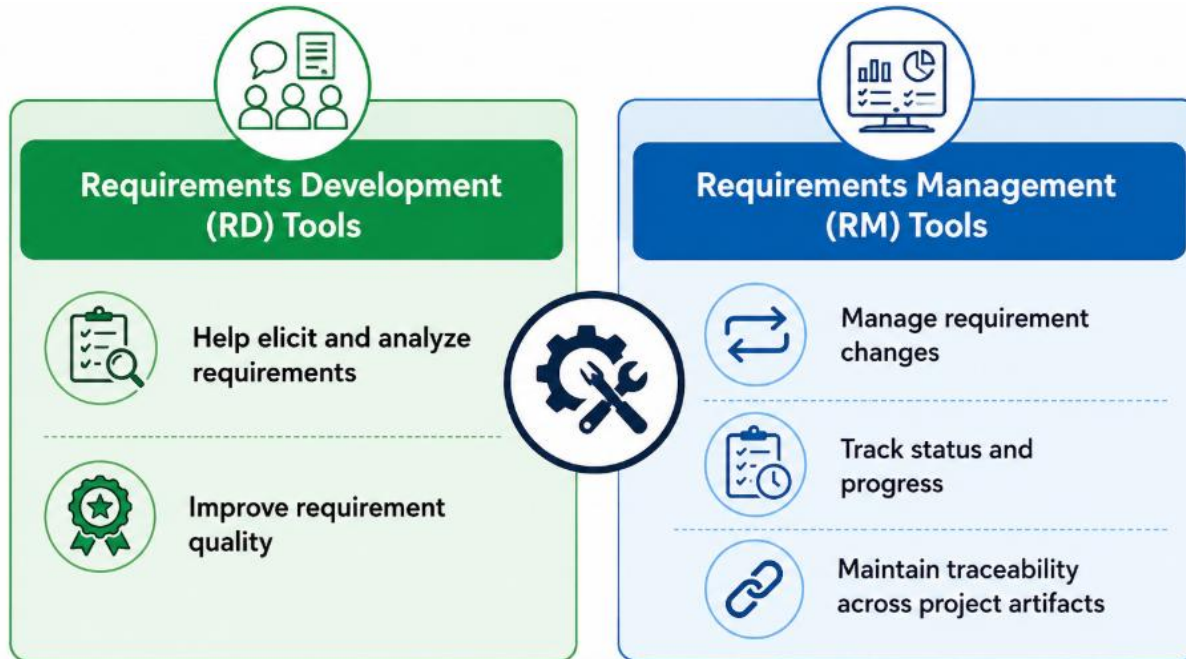
No easy storage for rejected or deleted requirements



Difficult to identify missing, duplicate, or unnecessary requirements

# Requirements Engineering Tools

A software applications that help teams systematically collect, document, analyze, and track project requirements



# RD Tools



Used by Business Analysts (BAs) to elicit and document requirements.



Improve efficiency and stakeholder collaboration.

Support different communication styles:



Textual



Visual



Audio



Main categories:



## Elicitation Tools

Help gather requirements from stakeholders using techniques like interviews, surveys, brainstorming, and workshops.



## Prototyping Tools

Help create mockups or prototypes to visualize requirements and validate ideas early with stakeholders.



## Modeling Tools

Help represent and analyze requirements using models like use cases, process flows, data models, and diagrams.

# RD – Elicitation Tools

**Purpose:** Gather and organize requirements effectively.

**Features:**

- ✓ Note-taking during elicitation sessions
- ✓ Mind mapping for brainstorming
- ✓ Audio recording and playback
- ✓ Linking audio recordings to written notes
- ✓ Requirement quality checks (ambiguity detection)
- ✓ Automatic diagram generation from text
- ✓ Collaborative voting for requirement prioritization

# RD – Prototyping Tools



**Purpose:** Create visual representations of systems.

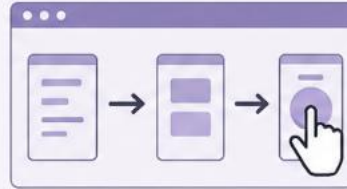
## Types



Low-fidelity wireframes



Screen mock-ups



Interactive simulations

## Benefits



Visualize system behavior



Collect user feedback early



Support navigation and workflow testing

## Advanced Features



Version control



Feedback management



Requirements linking



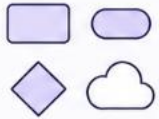
Code generation

# RD – Modeling Tools



**Purpose:** Create structured requirement models and diagrams.

## Capabilities



Standard shapes, notations, and syntax

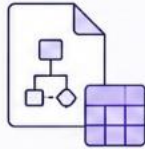
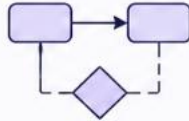
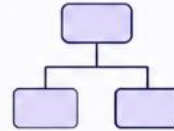


Diagram templates and examples



Automatic shape connections



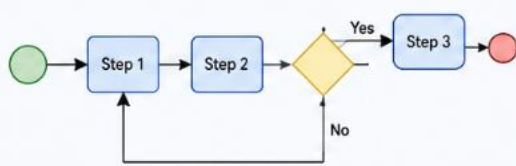
Cleaner and more consistent diagrams



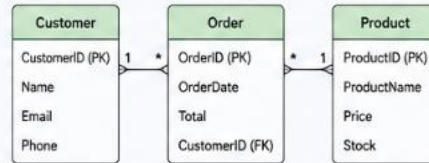
Easier diagram updates through drag-and-drop editing

## Examples

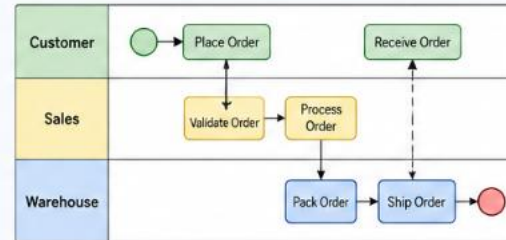
### Process diagrams



### Data models



### Swimlane diagrams



# RD – Modeling Tools ... Cont'd

## Benefits

- ✓ Better visualization of requirements
- ✓ Improved consistency and traceability
- ✓ Easier collaboration and analysis
- ✓ Link requirements to model elements

## Limitations

- Cannot detect all missing requirements
- Cannot determine logical correctness
- Do not replace critical thinking or peer reviews

# RM Tools



**Purpose:** Manage, track, and control requirements throughout the project lifecycle.



Store requirements in a multiuser database



Overcome limitations of document-based management

## Support:



Requirement storage



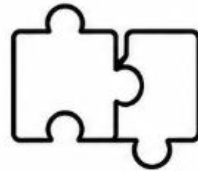
Traceability



Change management



Reporting



Often integrated with Requirements Development (RD) tools

# Why Use RM Tools?



More mature  
and widely  
available than  
RD tools



Help maintain  
control of  
requirements as  
projects evolve



Serve as a  
central repository  
for all  
stakeholders

## Especially Valuable When:



Projects  
are large



Requirements  
change frequently



Team members need  
consistent information

# Version & Change Management

RM tools help teams:



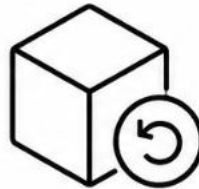
Create and manage  
requirement baselines



Record reasons  
for changes



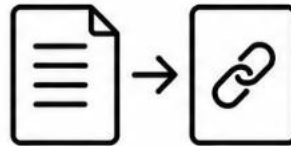
Track requirement  
versions



Restore previous  
versions if needed

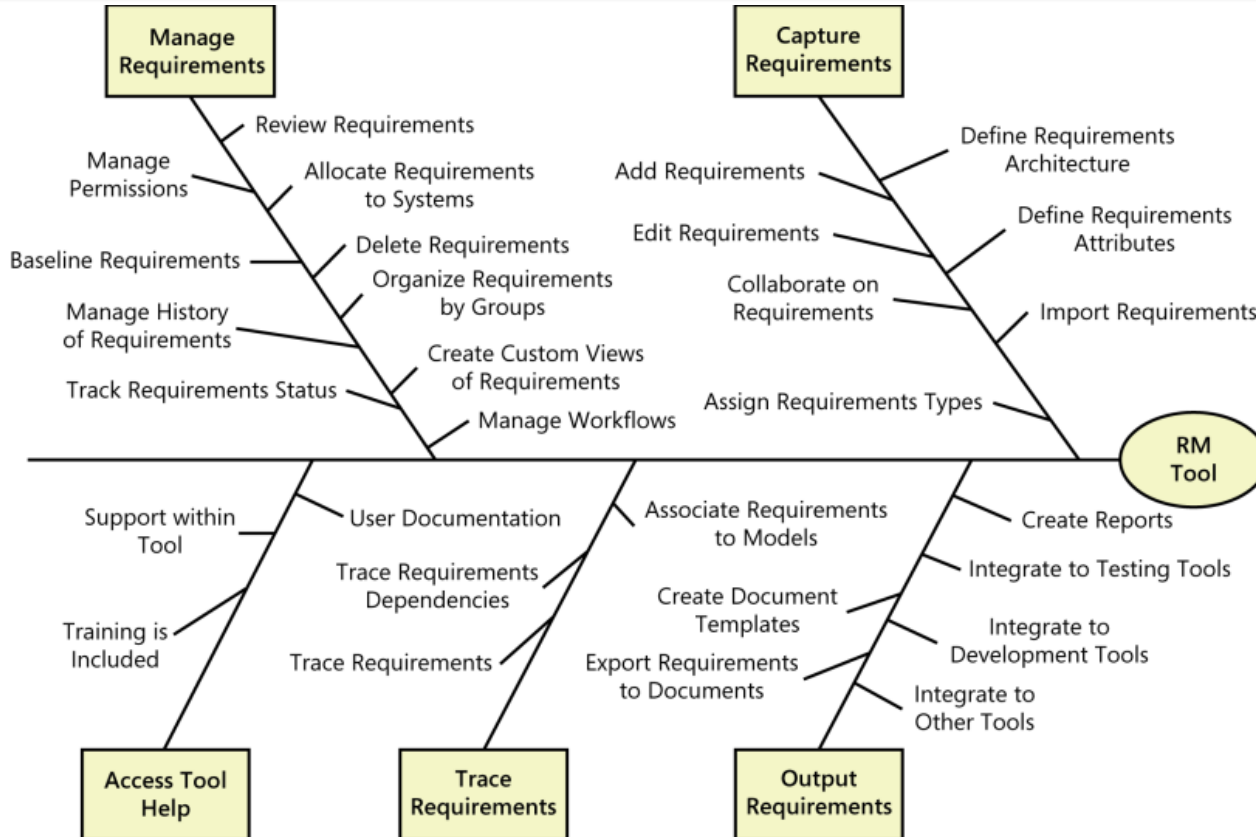


Maintain  
change history



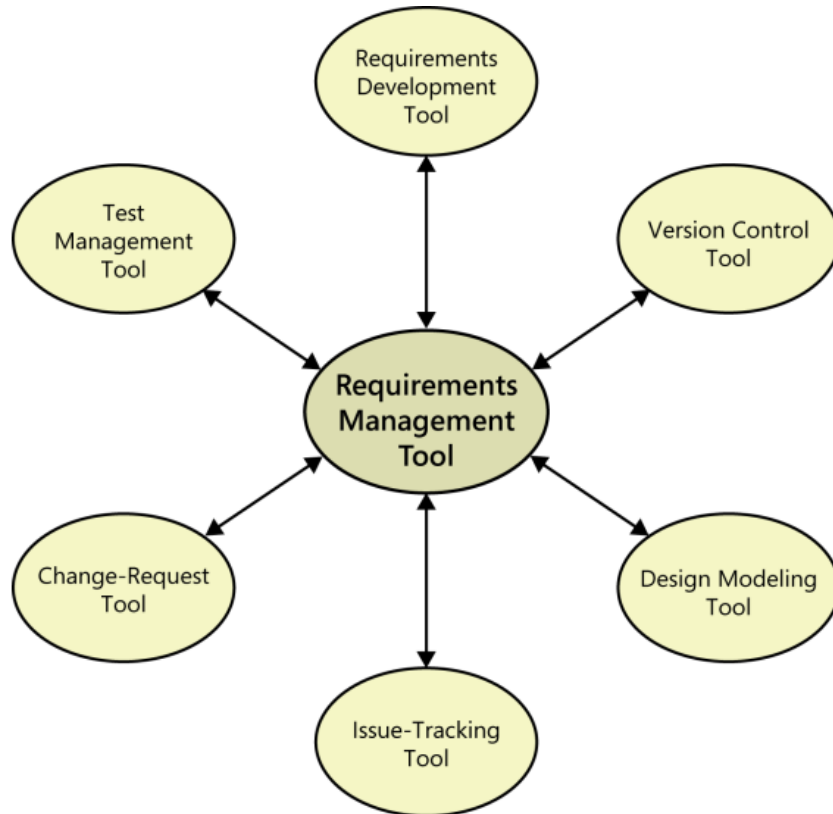
Link change requests  
to affected  
requirements

# RM Tools – Capabilities



**Figure 2:** Common RM tool features  
(Source: Beatty, 2013)

# RM Tools – Integration with Other Tools



**Figure 3:** RM tools integrate with other kinds of software tools

(Source: Beatty, 2013)

# Agile Requirements Engineering



Iterative requirements process



Continuous stakeholder involvement



Adaptive planning



Incremental delivery



Embraces change

The practice of **discovering**, **documenting**, **prioritizing**, and **managing** requirements within an **agile** development **environment**.

# Agile Manifesto and Requirements



Individuals and interactions

**Over**

Processes and Tools



Working software

**Over**

Extensive Documentation



Customer collaboration

**Over**

Contract Negotiation



Responding to change

**Over**

Following a  
rigid plan

# Agile Principles in RE



Customer  
Satisfactions



Changing  
Requirements



Frequent  
Delivery



Communicate  
Regularly



Face-to-face  
communication



Measure work  
progress



Development  
process



Good  
Design



Measure  
progress



Support team  
member

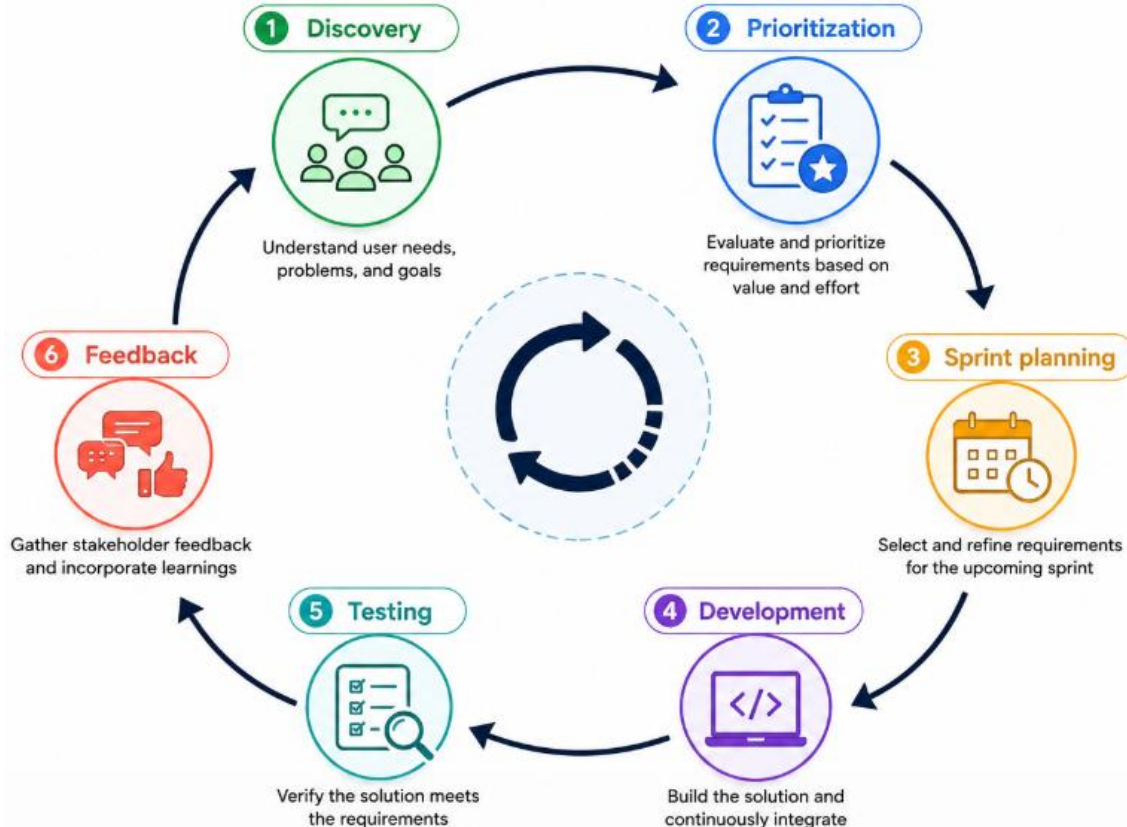


Continue  
Seeking Result



Reflect and  
adjust regularly

# Agile Requirements Lifecycle



# Agile Requirements & Backlogs

## Requirements Backlog



Central repository



Prioritized requirements



Dynamic list



Continuously updated

## Product Backlog



Managed by Product Owner



Contains all requirements



Prioritized by value



Continuously refined



## Sprint Backlog



Selected backlog items



Current sprint work



Team commitment



Sprint goals

# User stories



## User Stories



Lightweight requirement



User-focused



Business value



Easy to understand

- ✓ A short, simple, and informal description of a software feature written from the perspective of an end-user.
- ✓ It describes **who**, **what** and **why**.



## User Story Template

As a [User]



I want [Function]



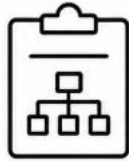
So that [Benefit]



# User Stories ... Cont'd

## Characteristics (INVEST)

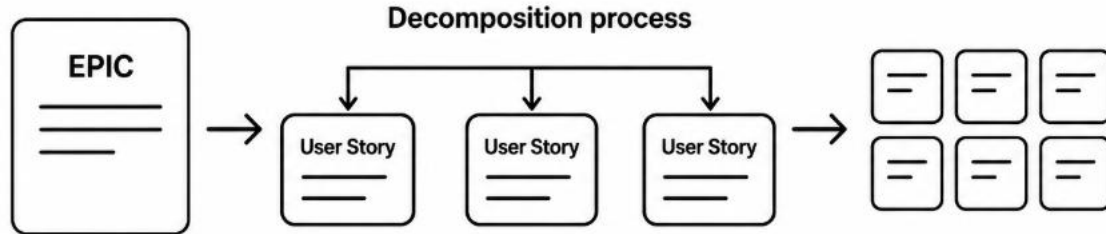
- ✓ Independent
- ✓ Negotiable
- ✓ Valuable
- ✓ Estimable
- ✓ Small
- ✓ Testable



**Epic** = Large requirement



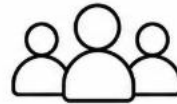
**User Story** = Smaller requirement



Incremental delivery



Easier management



# Figma & Balsamiq for Requirement Prototyping



Create visual system prototypes



Improve stakeholder understanding



Support interface design discussions



Detect usability issues early



Encourage feedback before coding

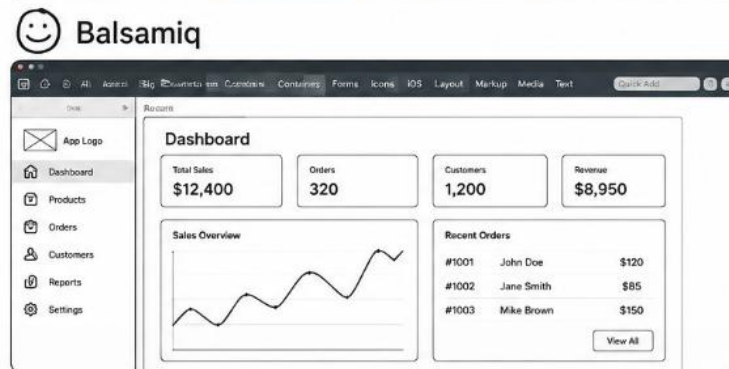
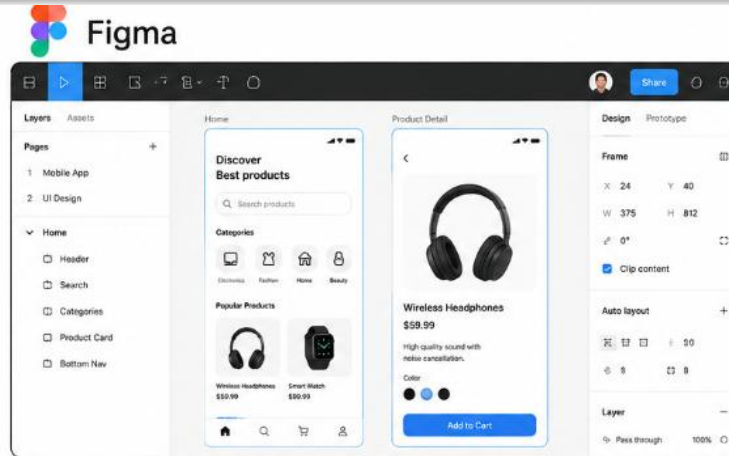


Figure 4. *Figma and Balsamiq for Requirement Prototyping*  
**Note.** Image generated using ChatGPT by OpenAI (2026).

# Challenges of Agile Requirements



Scope changes



Stakeholder availability



Incomplete requirements



Traceability complexity



Scaling challenges

# Best Practices to Address Challenges



**Maintain active stakeholder engagement**



**Refine backlog regularly**



**Define clear acceptance criteria**



**Establish requirement traceability**



**Use appropriate RE tools**



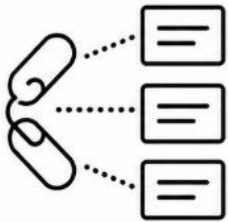
**Conduct frequent reviews and retrospectives**



**Balance flexibility with governance**

# Traceability Tools

- ✓ Although agile projects emphasize lightweight documentation, traceability remains extremely important.
- ✓ The ability to track relationships among **requirements**, **design elements**, **code modules**, **test cases**, and **business objectives**



Automated  
linkage



Change  
impact  
analysis



Coverage  
tracking



Audit  
support



Reporting  
capabilities

# Common Requirements Engineering Tools

## Traditional Requirements Management Tools



These are often used in large, regulated, and documentation-intensive projects.



IBM DOORS



Jama Connect



Polarion ALM



codeBeamer



Helix RM

**Best for:**



Banking systems



Aerospace projects



Government projects



Healthcare systems




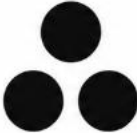



Safety-critical systems

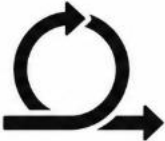



# Common Requirements Engineering Tools

## Agile Requirements Management Tools

These focus on backlogs, user stories, sprint planning, and collaboration.

Jira	Azure DevOps	Trello	Asana	Monday.com
				

Best for:

			
Scrum projects	Kanban projects	Startups	Continuous delivery environments

# Common Requirements Engineering Tools

## Modeling and Analysis Tools

Used for UML, business process modeling, and requirement analysis.

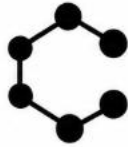
Enterprise Architect



Microsoft Visio



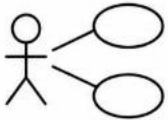
Lucidchart



Visual Paradigm



Used for:



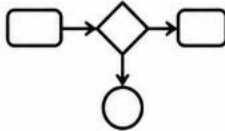
Use Case Diagrams



Activity Diagrams



Class Diagrams



Business Process Models



Requirement Models

# Common Requirements Engineering Tools

## Collaboration and Documentation Tools

Support communication and requirement sharing.

Confluence



Microsoft Teams



SharePoint



Google Docs



Used for:



Requirement workshops



Stakeholder communication



Requirement reviews



Documentation repositories

# Common Requirements Engineering Tools

RE Activity	Purpose	Example Tools	Agile Support	Traditional Support
<b>Requirements Elicitation</b>	Capture stakeholder needs	Microsoft Teams, Zoom, Miro	✓	✓
<b>Requirements Documentation</b>	Document requirements	Microsoft Word, Confluence	✓	✓
<b>Requirements Modeling</b>	Create diagrams and models	Microsoft Visio, Lucidchart, Enterprise Architect	✓	✓

# Common Requirements Engineering Tools

<b>RE Activity</b>	<b>Purpose</b>	<b>Example Tools</b>	<b>Agile Support</b>	<b>Traditional Support</b>
<b>Requirements Management</b>	Manage and organize requirements	IBM DOORS, Jama Connect	✓	✓
<b>Traceability Management</b>	Maintain requirement links	IBM DOORS, Jama Connect	✓	✓
<b>Change Management</b>	Control requirement changes	Azure DevOps, Jira	✓	✓
<b>Testing &amp; Validation</b>	Verify requirements	TestRail, HP ALM	✓	✓

# Summary



## 1 Agile embraces change

Agile methodologies welcome change, even late in development, to deliver more value to customers.



## 2 User stories drive requirements

User stories capture user needs in simple language and help ensure the product delivers value.



## 3 Backlogs manage work

Product and sprint backlogs prioritize and organize work, ensuring the team focuses on what matters most.



## 4 Tools improve efficiency

Agile and requirements management tools enhance collaboration, visibility, and productivity.



## 5 Traceability remains important

Maintaining traceability ensures alignment from requirements to delivery and supports quality and compliance.

# References

1. Beatty, K. W. (2013). Software Requirements (3rd ed.). Washington: Microsoft Press.

# Thank You!

