

1. UML Use Case Diagrams

Depicts:

- Actors (users or external systems)
- Use cases (services or functions the system provides)
- Relationships between actors and use cases

Purpose:

To capture *functional requirements* from the user's point of view and show how external entities interact with the system.

2. UML Robustness Diagrams

Depicts:

- Boundary objects (interfaces)
- Control objects (logic/behavior coordinators)
- Entity objects (data and business entities)
- Interactions among them

Purpose:

To bridge **use cases** and **detailed design**, ensuring the scenario logic is consistent and that responsibilities are well assigned before writing sequence diagrams or code.

3. UML Activity Diagrams

Depicts:

- Workflows and processes
- Actions, decisions, merges, forks, joins
- Parallel and sequential flows

Purpose:

To model **business processes** or **algorithmic workflows**, showing how activities proceed and branch over time.

4. UML Class Diagrams

Depicts:

- Classes, attributes, operations
- Relationships (association, aggregation, composition, inheritance)

Purpose:

To model the **static structure** of the system and define its object-oriented architecture.

5. UML Object Diagrams

Depicts:

- Concrete instances of classes (objects)
- Specific values in attributes
- Links (instances of associations)

Purpose:

To show a **snapshot of the system at a particular moment**, useful for illustrating examples or debugging relationships.

6. UML Package Diagrams

Depicts:

- Packages (groupings of classes or subsystems)
- Dependencies between packages

Purpose:

To represent the **high-level organization** of the system and manage complexity by grouping related model elements.

7. UML Sequence Diagrams

Depicts:

- Objects participating in a scenario
- Messages passed between them over time
- Lifelines and activation bars

Purpose:

To detail **how a specific use case plays out** through time-ordered interactions.

8. UML Communication Diagrams (Collaboration Diagrams)

Depicts:

- Objects participating in a scenario
- Links between objects
- Numbered messages (ordered but not time-scaled like sequence diagrams)

Purpose:

To emphasize **object relationships** and how they collaborate to accomplish behavior.

9. UML State Diagrams

Depicts:

- States an object can be in
- Transitions triggered by events
- Entry/exit actions

Purpose:

To model the **lifecycles of stateful objects**, showing how they transition in response to events.

10. UML Component Diagrams

Depicts:

- Software components (modules, libraries, executables)
- Interfaces they require or provide
- Dependencies between components

Purpose:

To show the **physical or logical modular structure** of the system at a high level.

11. UML Deployment Diagrams

Depicts:

- Hardware nodes (servers, devices)
- Software artifacts deployed on those nodes
- Communication paths among nodes

Purpose:

To illustrate the **physical architecture** of the running system—how and where software is deployed.