UML & Tool Introduction

Context

- Introduction of the Project and Rolles
- UML Introduction
- UML Diagram Types
- UML Tool Introduction
- About Vision and Project Plan Templates (with OpenUp)

About the Project

- Sports Center Membership System (SCMS)

UML

- The Unified Modeling Language (UML) is a graphical notation for drawing diagrams of software concepts.
- A language for specifying, visualizing, constructing, and documenting the artifacts of software systems
- *Conceptual (a problem domain)*, <u>*Specification (a proposed software design)*, and *Implementation (software implementation)*</u>

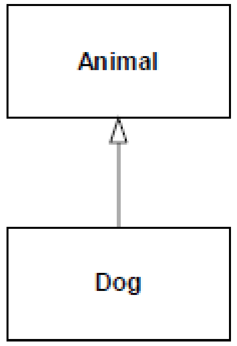


Figure 1-1
A Dog is an Animal

```
public class Animal {}
Public class Dog extends Animal {}
```

UML – The Basics

- Abstraction (A simplification or model of a complex concept, process, or real-World object – Help people understand something at an appropriate level)
- Encapsulation (Higlight the important aspects of an object – Hide the cumbersome internal details of the object – make the system easier to understand and reuse – make a system more extendible)
- Entities (object, class) and relationships between objects

UML Diagram Types

- UML has three main kinds of diagrams
  - Static diagrams describe the unchanging logical structure of software elements by depicting classes, objects, and data structures; and the relationships that exist between them.
  - Dynamic diagrams show how software entities change during execution by depicting the flow of execution, or the way entities change state.
  - Physical diagrams show the unchanging physical structure of software entities by depicting physical entities such as source files, libraries, binary files, data files, etc., and the relationships that exist between them.

UML Diagram Types

- The current UML standards call for 13 different types of diagrams
- These diagrams are organized into two distinct groups: structural diagrams and behavioral or interaction diagrams.

Structural UML diagrams

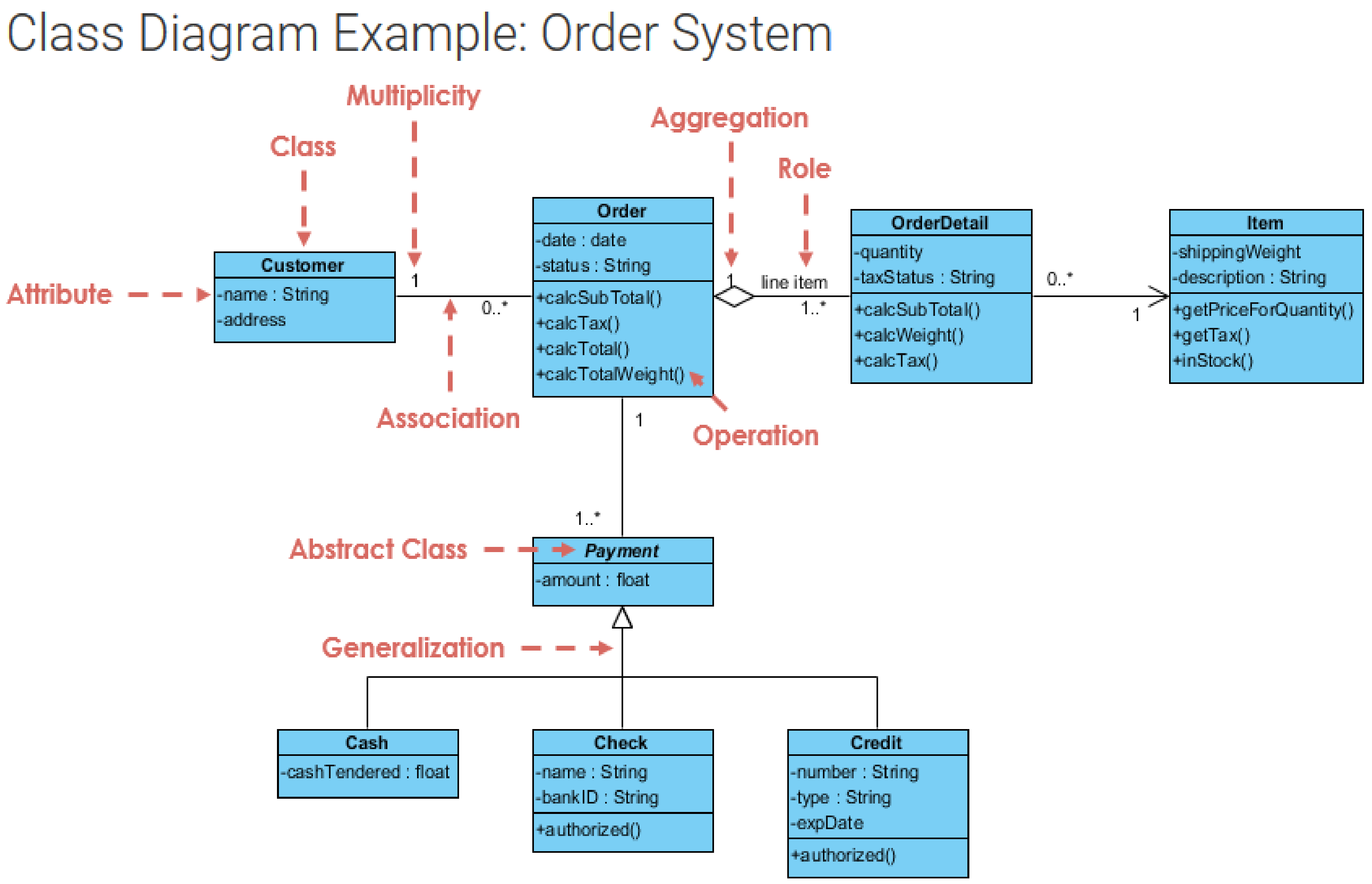
- **<u>Class diagram</u>**
- Package diagram
- Object diagram
- Component diagram
- Composite structure diagram
- Deployment diagram

Behavioral UML diagrams

- **<u>Activity diagram</u>**
- Sequence diagram
- **<u>Use case diagram</u>**
- State diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

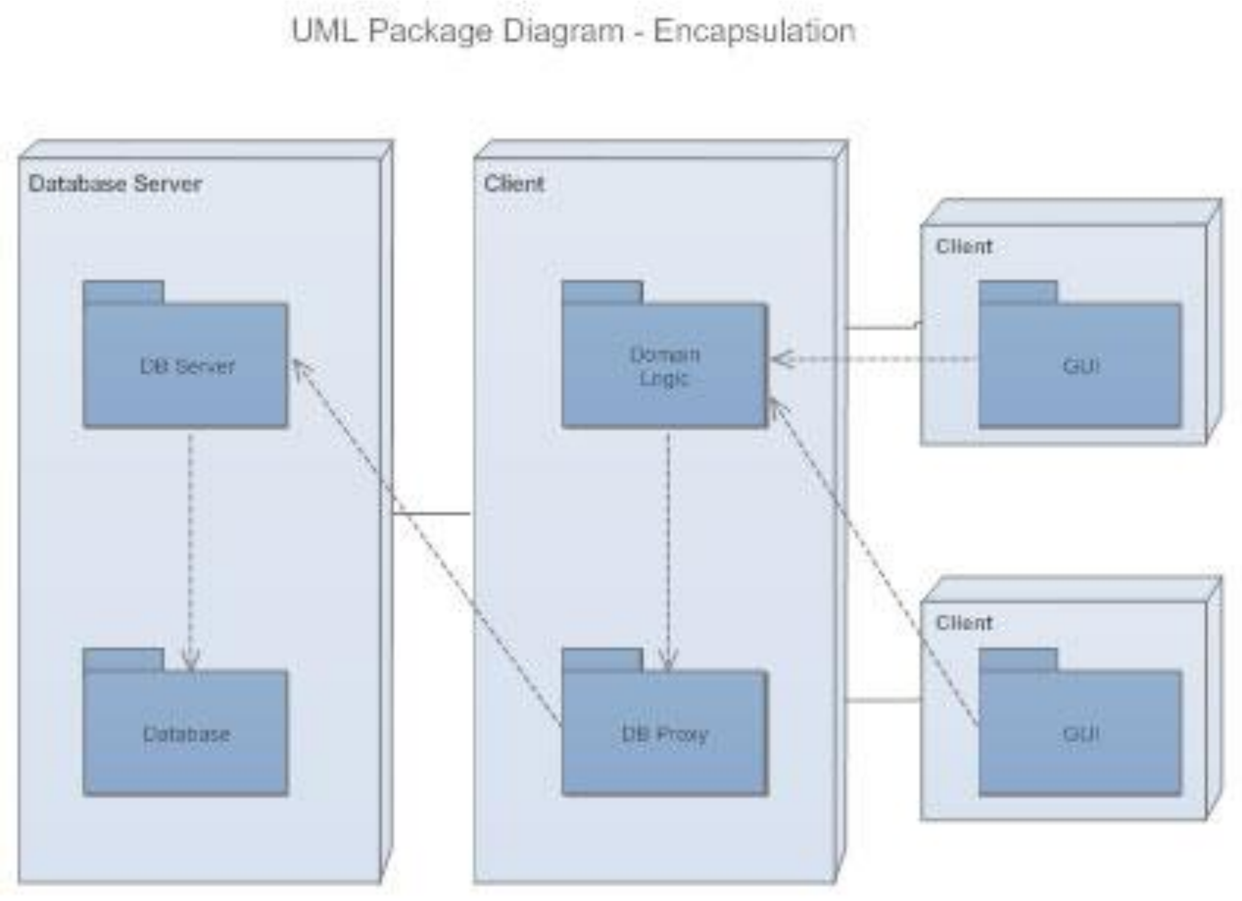
UML Diagram Types – Class Diagram

A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations.



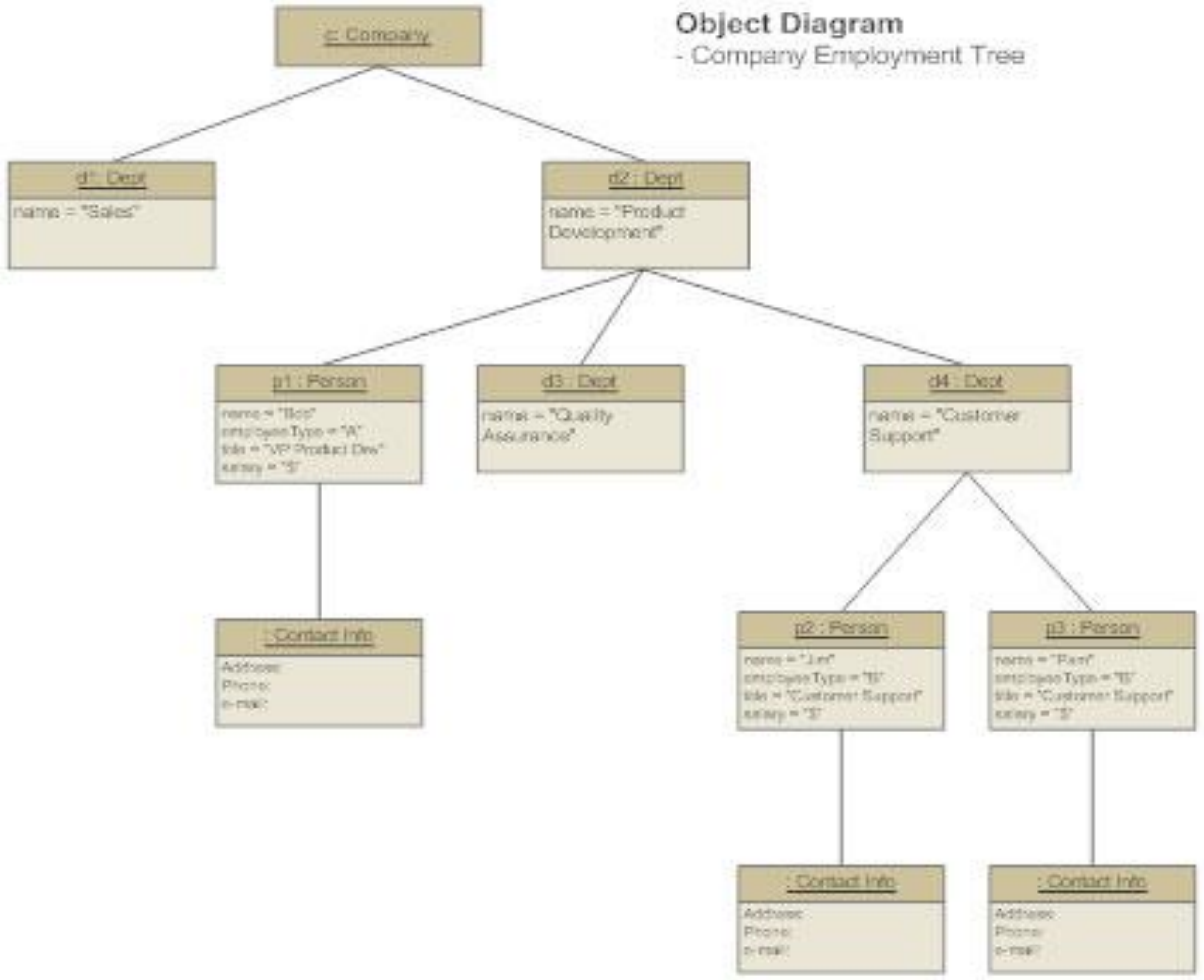
UML Diagram Types – Package Diagram

Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.



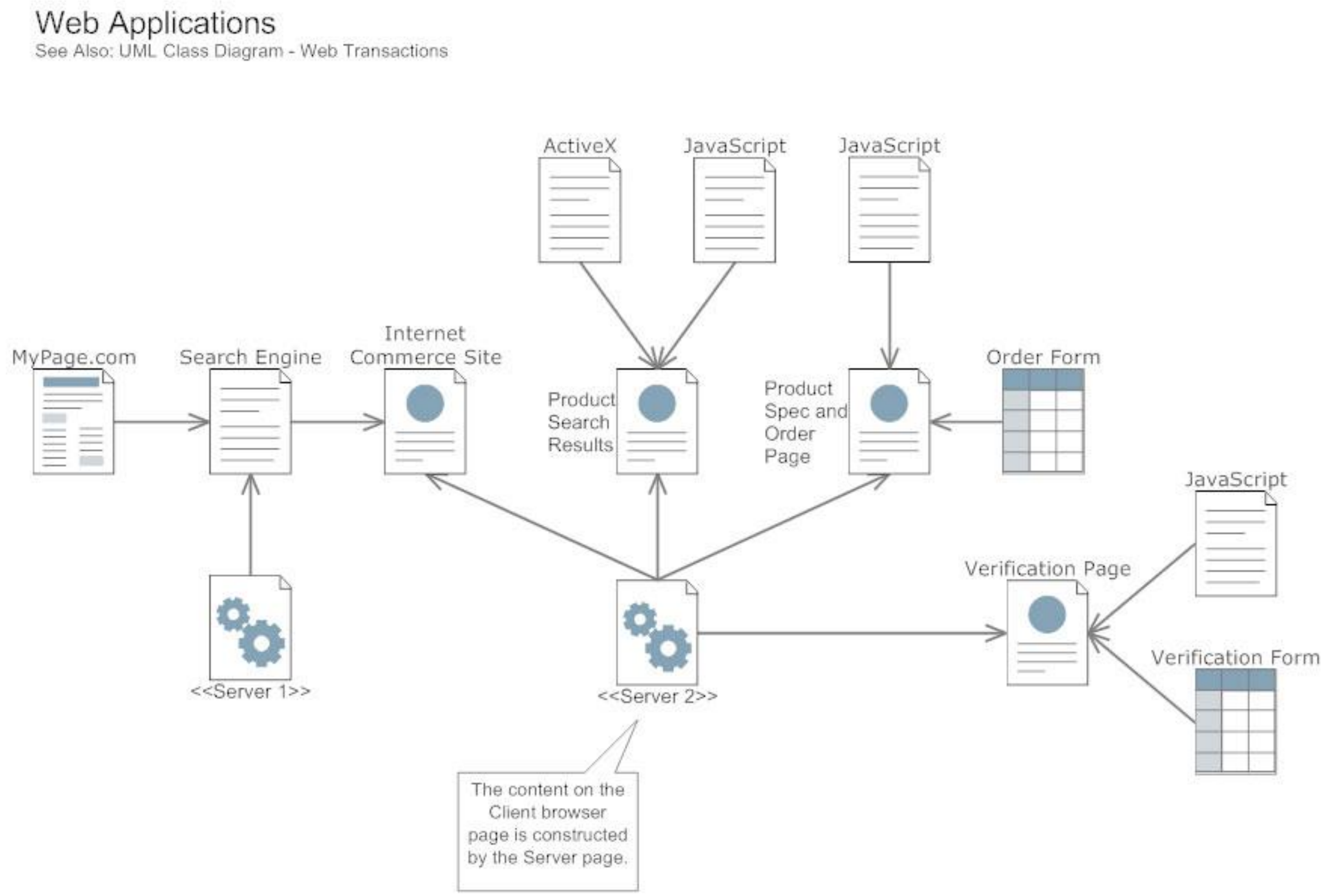
UML Diagram Types – Object Diagram

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.



UML Diagram Types – Component Diagram

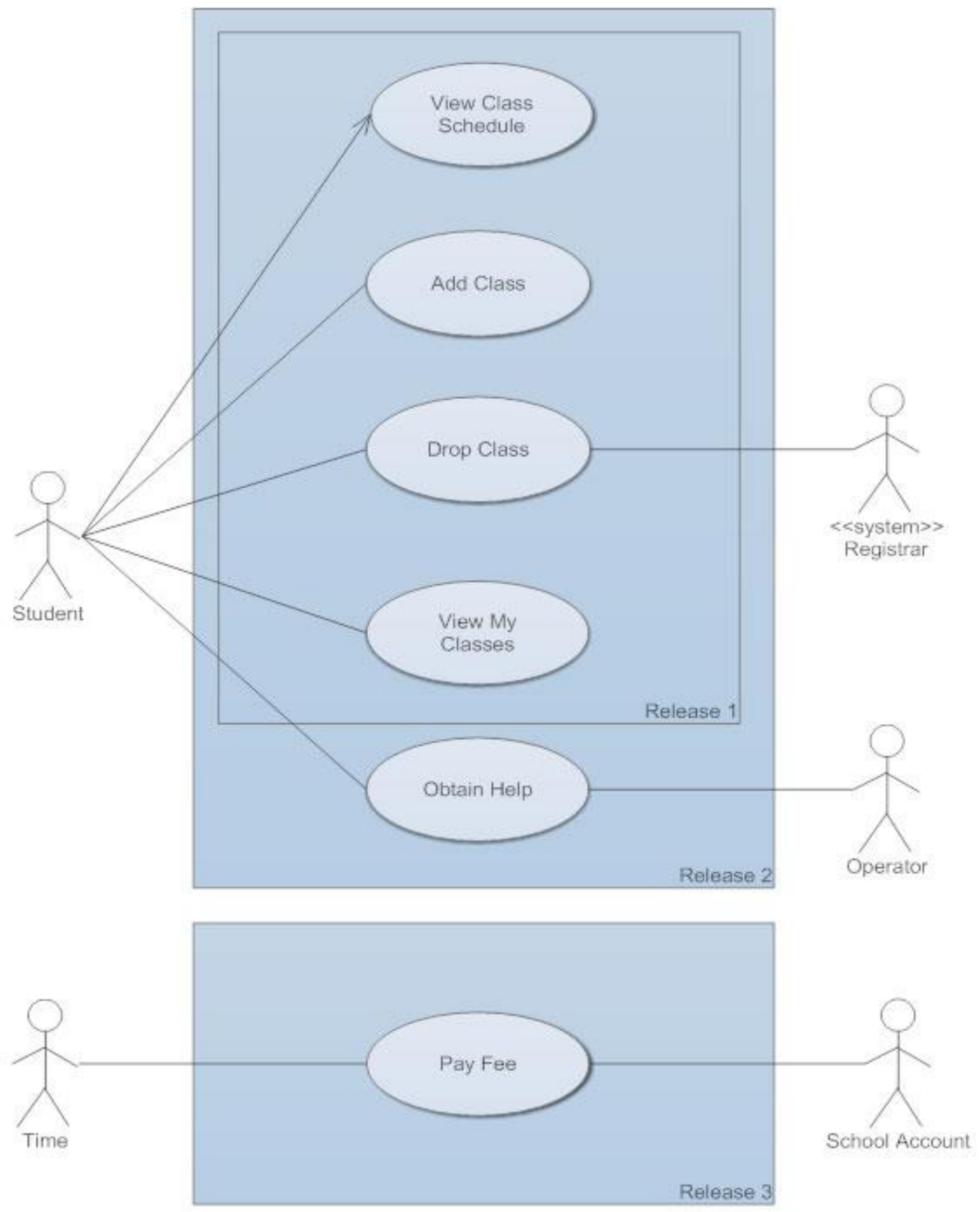
Component diagrams describe the organization of physical software components, including source code, run-time (binary) code, and executables.



UML Diagram Types – UseCase Diagram

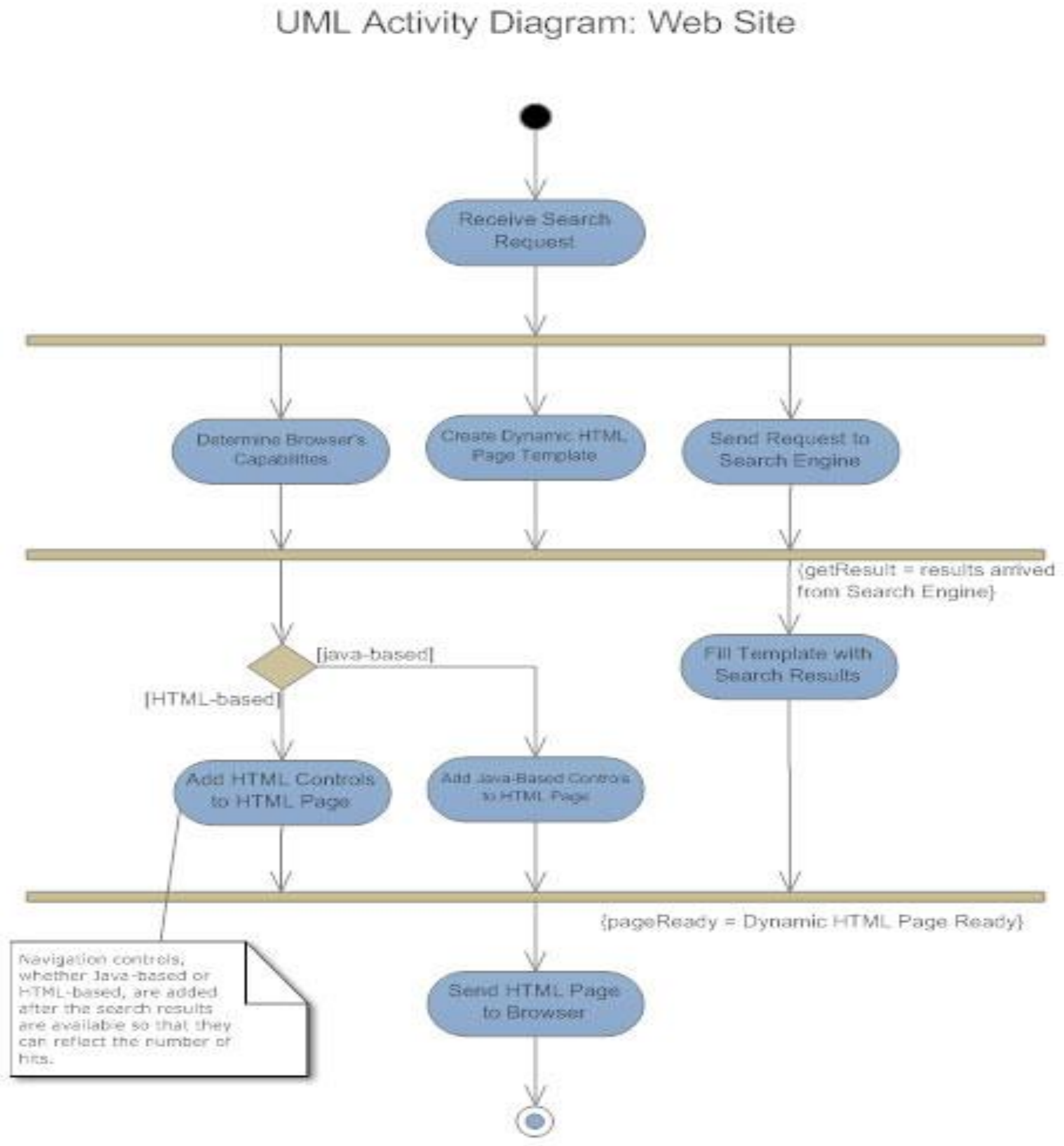
A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

Use Case Diagram: Class Registration



UML Diagram Types – Activity Diagram

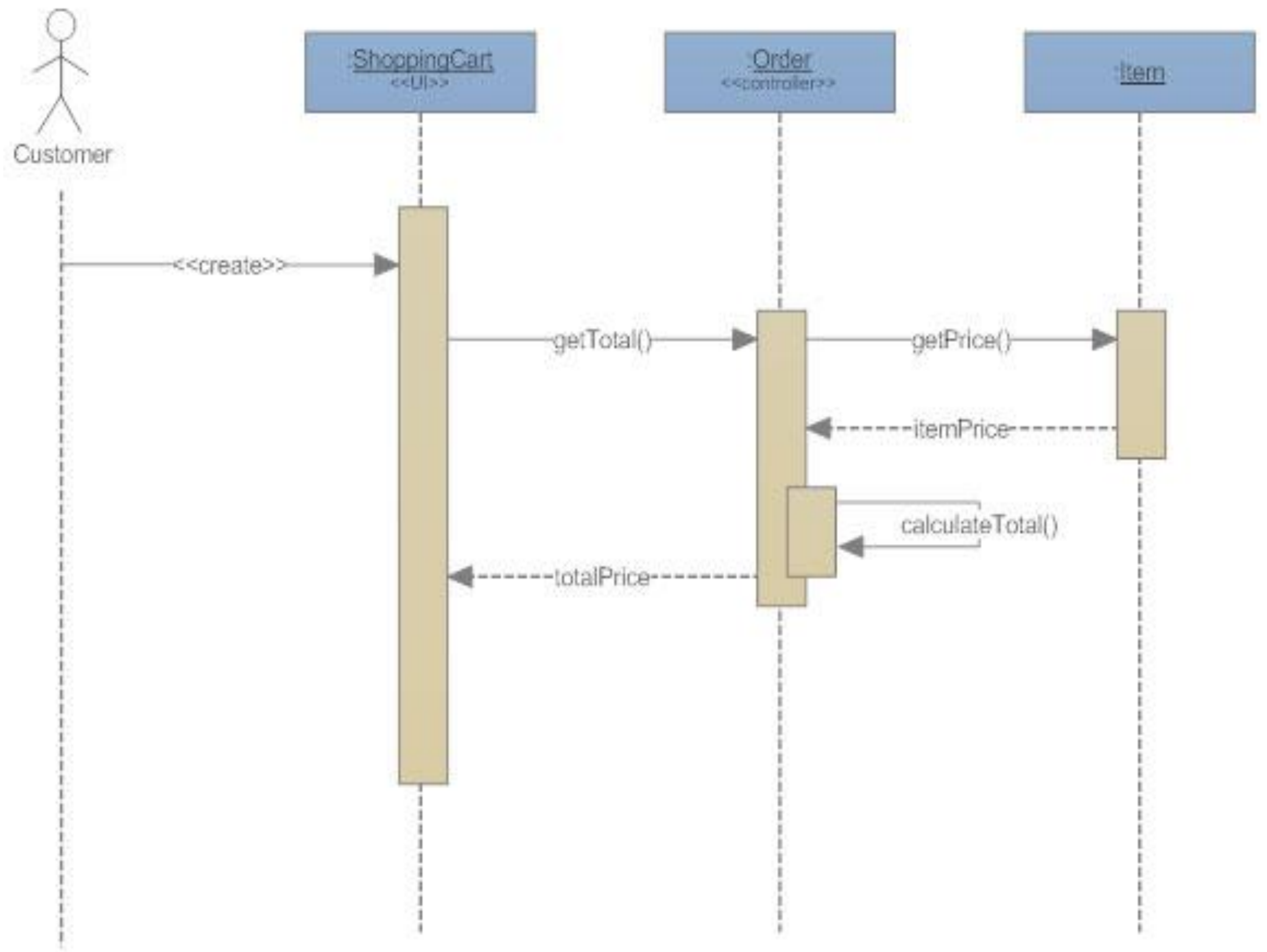
Activity diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation


UML Activity Diagram: Web Site
Receive Search Request
Determine Browser's Capabilities
Create Dynamic HTML Page Template
Send Request to Search Engine
{getResult = results arrived from Search Engine}
[java-based]
[HTML-based]
Fill Template with Search Results
Add HTML Controls to HTML Page
Add Java-Based Controls to HTML Page
{pageReady = Dynamic HTML Page Ready}
Navigation controls, whether Java-based or HTML-based, are added after the search results are available so that they can reflect the number of hits.
Send HTML Page to Browser

UML Diagram Types – Sequence Diagram

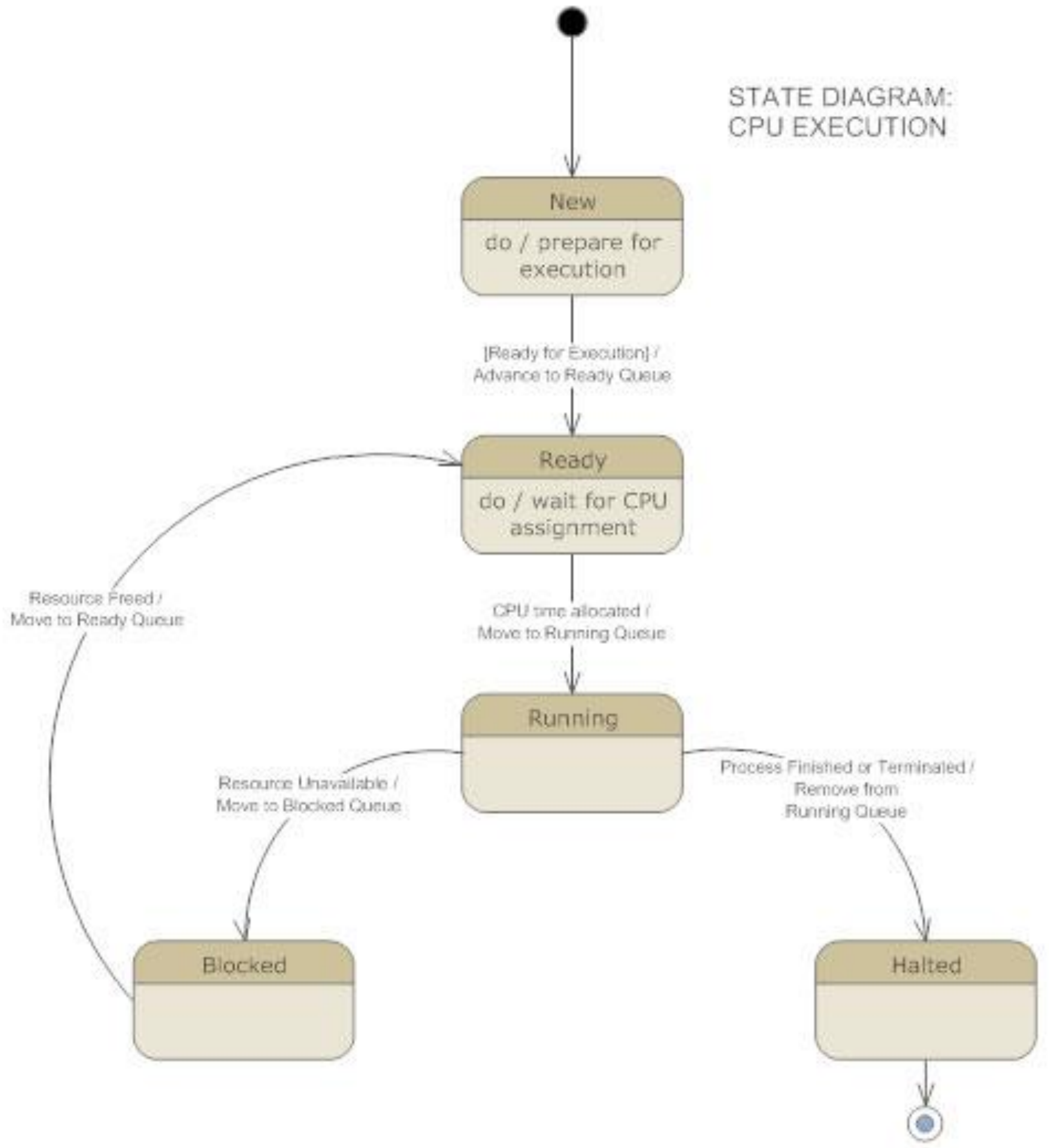
Sequence diagrams describe interactions among classes in terms of an exchange of messages over time

Sequence Diagram: Shopping Cart



UML Diagram Types – State Diagram

Statechart diagrams, now known as state machine diagrams and state diagrams describe the dynamic behavior of a system in response to external stimuli. State diagrams are especially useful in modeling reactive objects whose states are triggered by specific events


STATE DIAGRAM: CPU EXECUTION
New
do / prepare for execution
[Ready for Execution] / Advance to Ready Queue
Ready
do / wait for CPU assignment
CPU time allocated / Move to Running Queue
Running
Resource Unavailable / Move to Blocked Queue
Process Finished or Terminated / Remove from Running Queue
Blocked
Halted
Resource Freed / Move to Ready Queue

UML Tool

- Visual Paradigm Community Edition https://www.visual-paradigm.com/download/community.jsp
    - Visual Paradigm is a powerful, cross-platform and yet easy-to-use design and management tool for IT systems. Visual Paradigm provides software developers the cutting edge development platform to build quality applications faster, better and cheaper.
- SmartDraw https://www.smartdraw.com/
- MS Visio

Vision and Project Plan Templates (with OpenUp)

- Vision
    - http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/templates/resources/vision_tpl.dot
    - This artifact defines the view of the stakeholders of the technical solution to be developed. This definition is specified in terms of the key needs and features of the stakeholders. The vision contains an outline of the envisioned core requirements for the system.

Vision and Project Plan Templates (with OpenUp)

- Project Plan Template

  - http://epf.eclipse.org/wikis/openup/practice.mgmt.release_planning.base/guidances/templates/resources/project_plan_tpl.dot

  - A collaborative task that outlines an initial agreement on how the project will achieve its goals. The resulting project plan provides a summary-level overview of the project.

Product of the Lab. Project

- **System to be developed:**

Sports Center Membership System (SCMS)

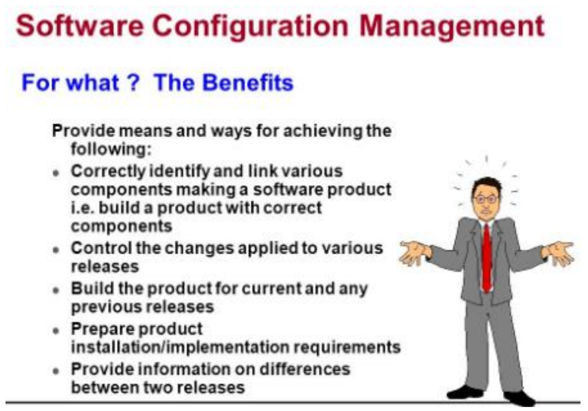
Specialized Team Roles

In addition to
Software Developer

Software Project Manager

Software Architect

Software Analyst



Software Configuration Management

For what ? The Benefits

Provide means and ways for achieving the following:

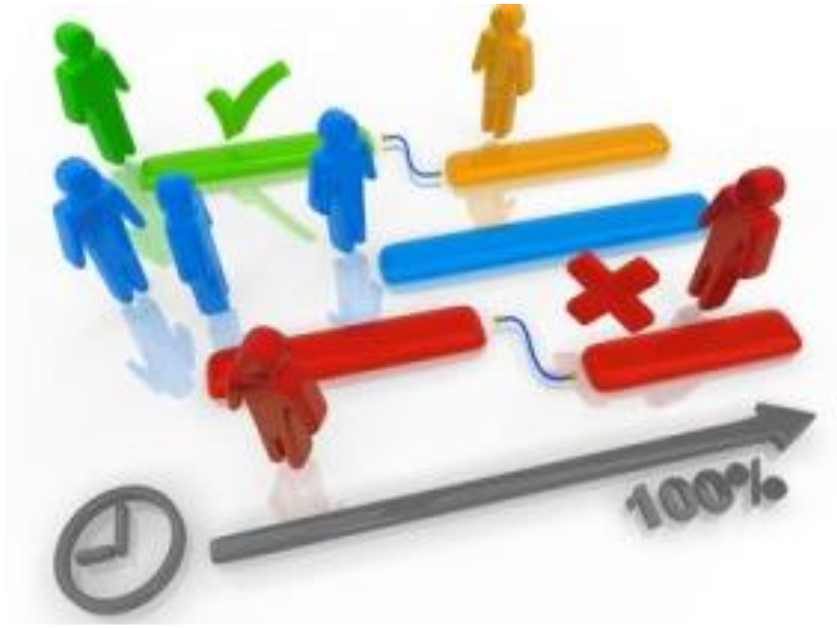
- Correctly identify and link various components making a software product i.e. build a product with correct components
- Control the changes applied to various releases
- Build the product for current and any previous releases
- Prepare product installation/implementation requirements
- Provide information on differences between two releases

Software Configuration Manager

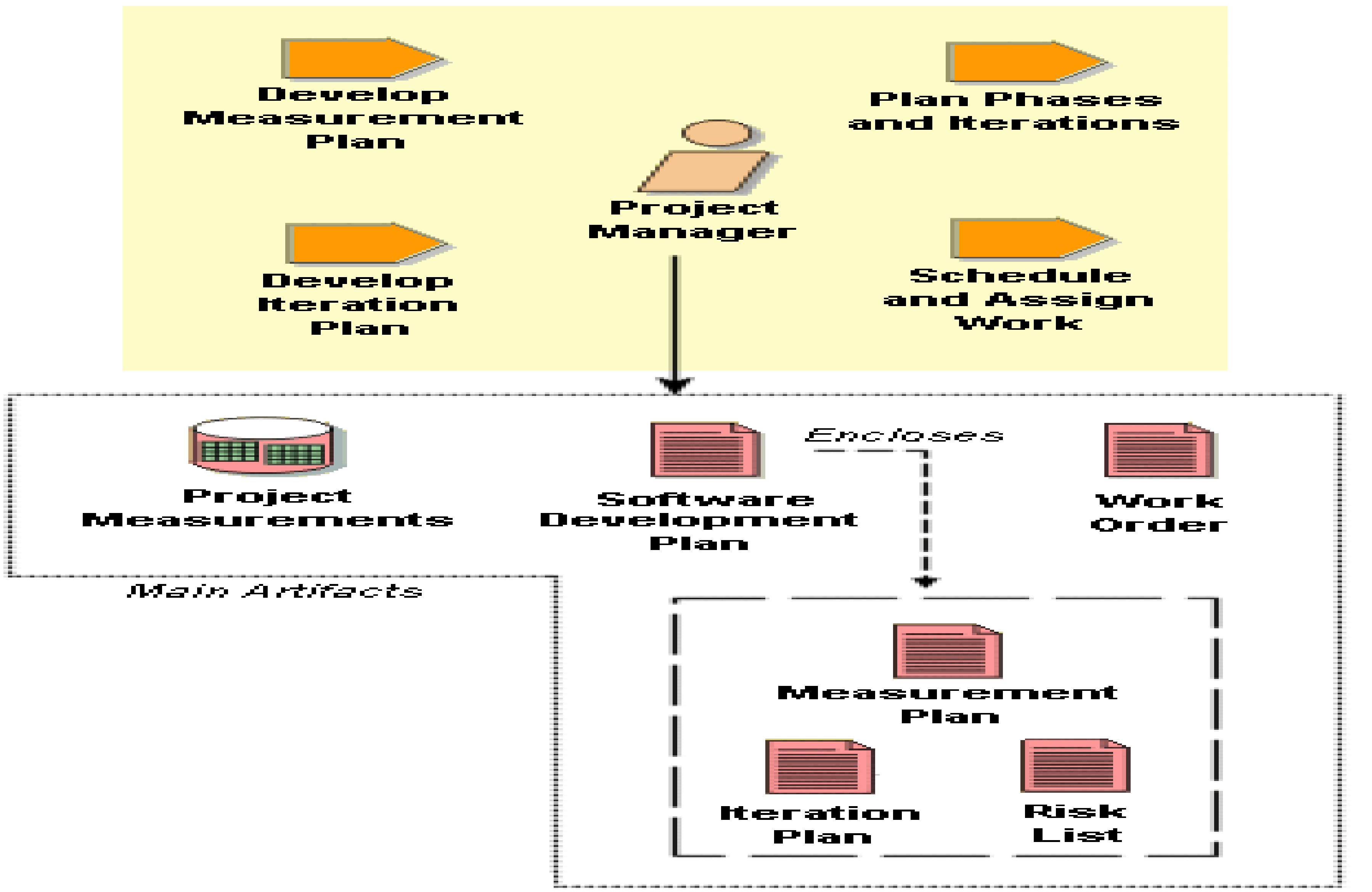
Software Tester

Project Manager

- The **project manager role** allocates resources, shapes priorities, coordinates interactions with customers and users, and generally keeps the project team focused on the right goal. The project manager also establishes a set of practices that ensure the integrity and quality of project artifacts.
- **<u>Responsibilities of a Project Manager</u>**
- **Planning** – scope, activity schedules, gantt chart, potential risks etc.
- **Setting goals -** For example: Complete the project within six months from start date in the budget of xxx amount.
- **Time Management**
- **Budget Allocation and Cost Estimates**
- **Implementation and Monitoring**



Project Manager

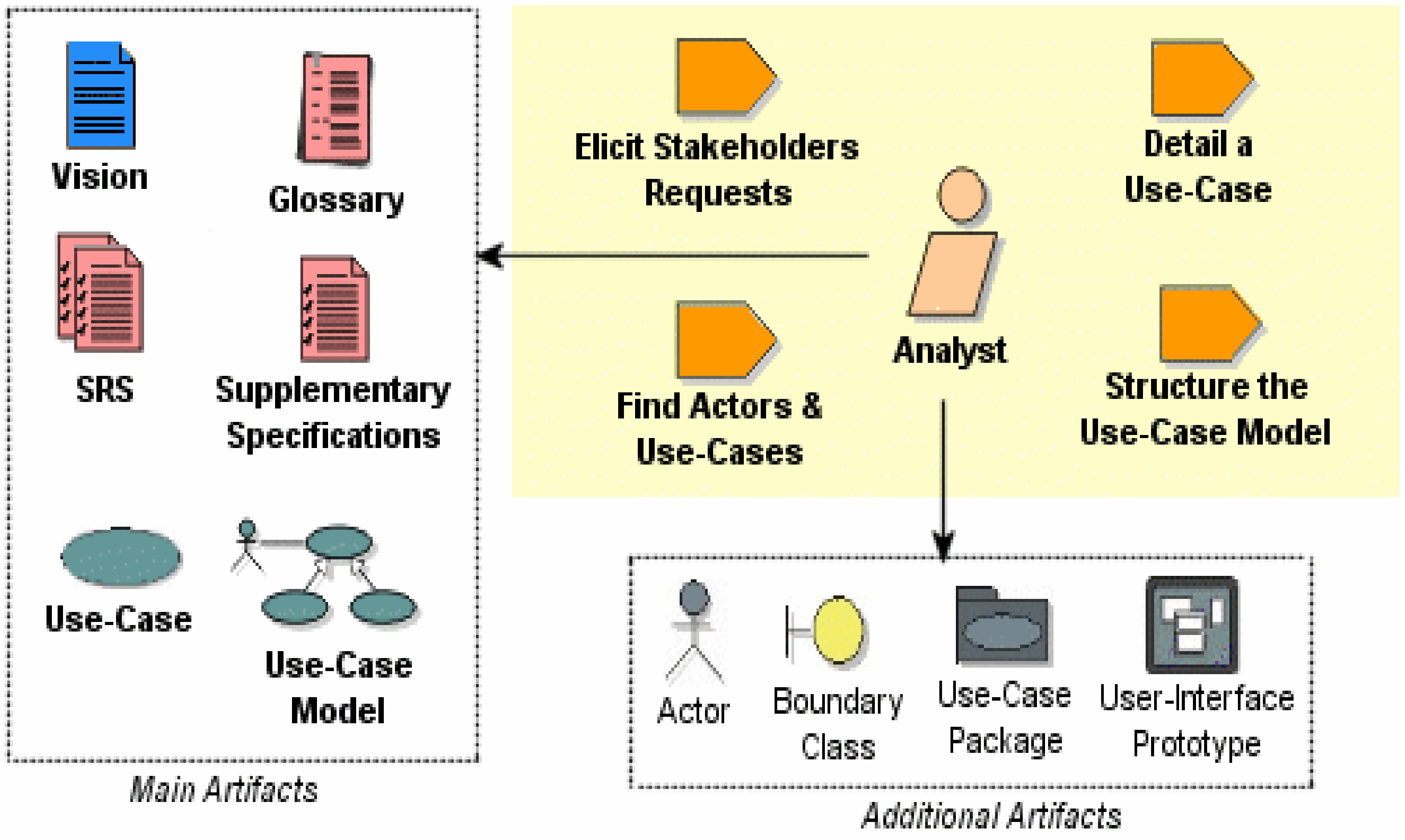


Software Analyst

J.KING Modern Analyst
I'm a new patient. I'm here to see Dr. Kingston.
Ah! You'll need to see our Healthcare Business Analyst first. The doctor needs somebody to blame for his failures just like the software developers!

- In a software development team, a **software analyst** is the person who studies the software application domain, prepares software requirements, and specification (Software Requirements Specification) documents.
  The software analyst is the seam between the software users and the software developers.
- S/he is responsible for
  - Performing complex analysis, designing and programming to meet business requirements.
  - Maintaining, managing and modifying all software systems and applications.
  - Defining specifications for complex software programming applications.
  - Interfacing with end-users and software consultants.
  - Developing, maintaining and managing systems, software tools and applications.
  - Resolving complex issues relating to business requirements and objectives.
  - Coordinating and supporting software professionals in installing and analyzing applications and tools.
  - Analyzing, developing and implementing testing procedures, programming and documentation.
  - Training and developing other software analysts.
  - Analyzing, designing and developing modifications and changes to existing systems to enhance performance.

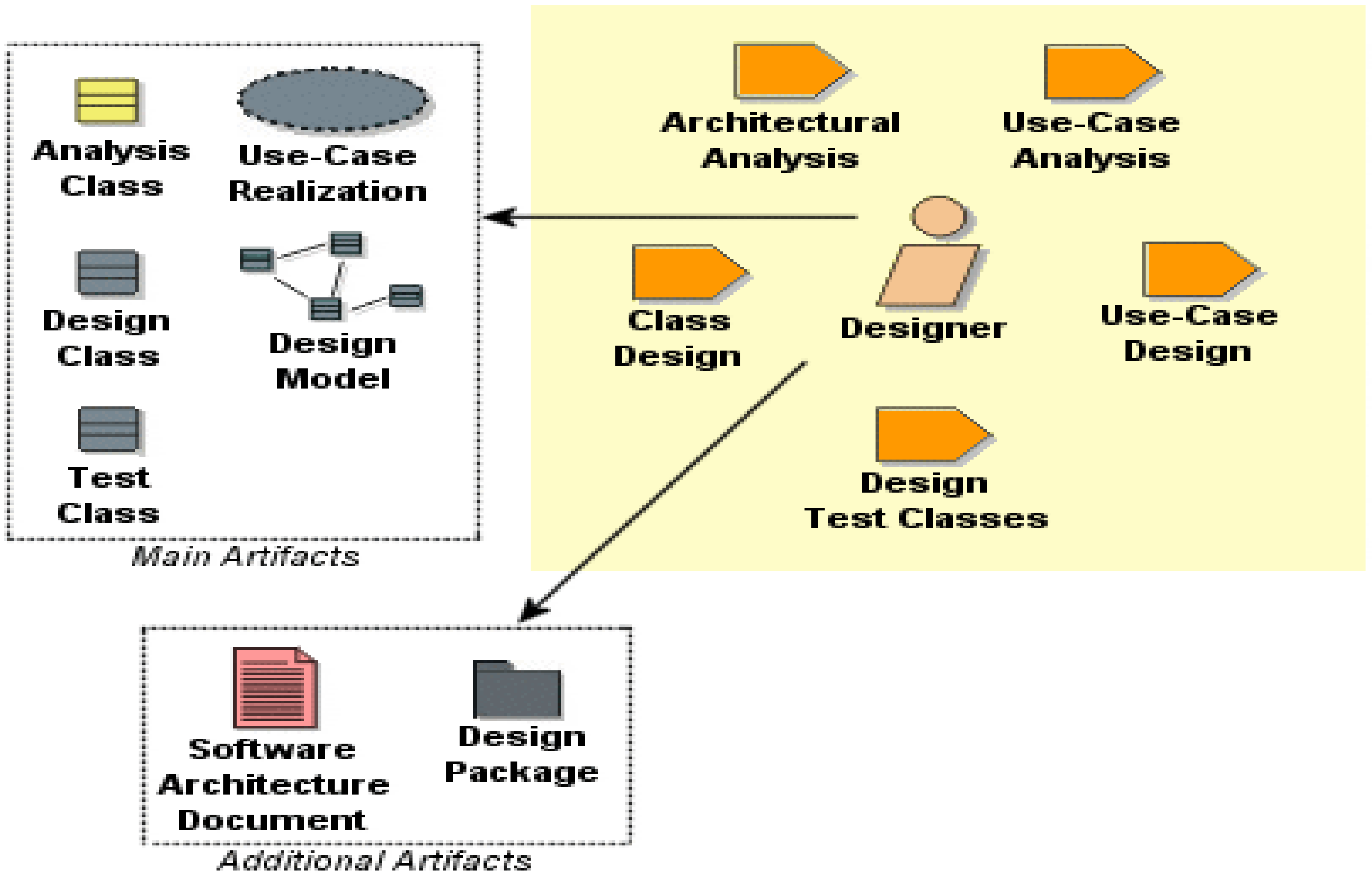
Software Analyst



Software Architect

- A **software architect/designer** is a software expert who makes high-level design choices and dictates technical standards, including software coding standards, tools, and platforms.
- Examples of Software Architect responsibilities
  - Design, develop and execute software solutions to address business issues.
  - Provide architectural blueprints and technical leadership to our IT team.
  - Evaluate and recommend tools, technologies and processes to ensure the highest quality product platform.
  - Collaborate with peer organizations, quality assurance and end users to produce cutting-edge software solutions.
  - Interpret business requirements to articulate the business needs to be addressed.
  - Troubleshoot code level problems quickly and efficiently.

Software Architect

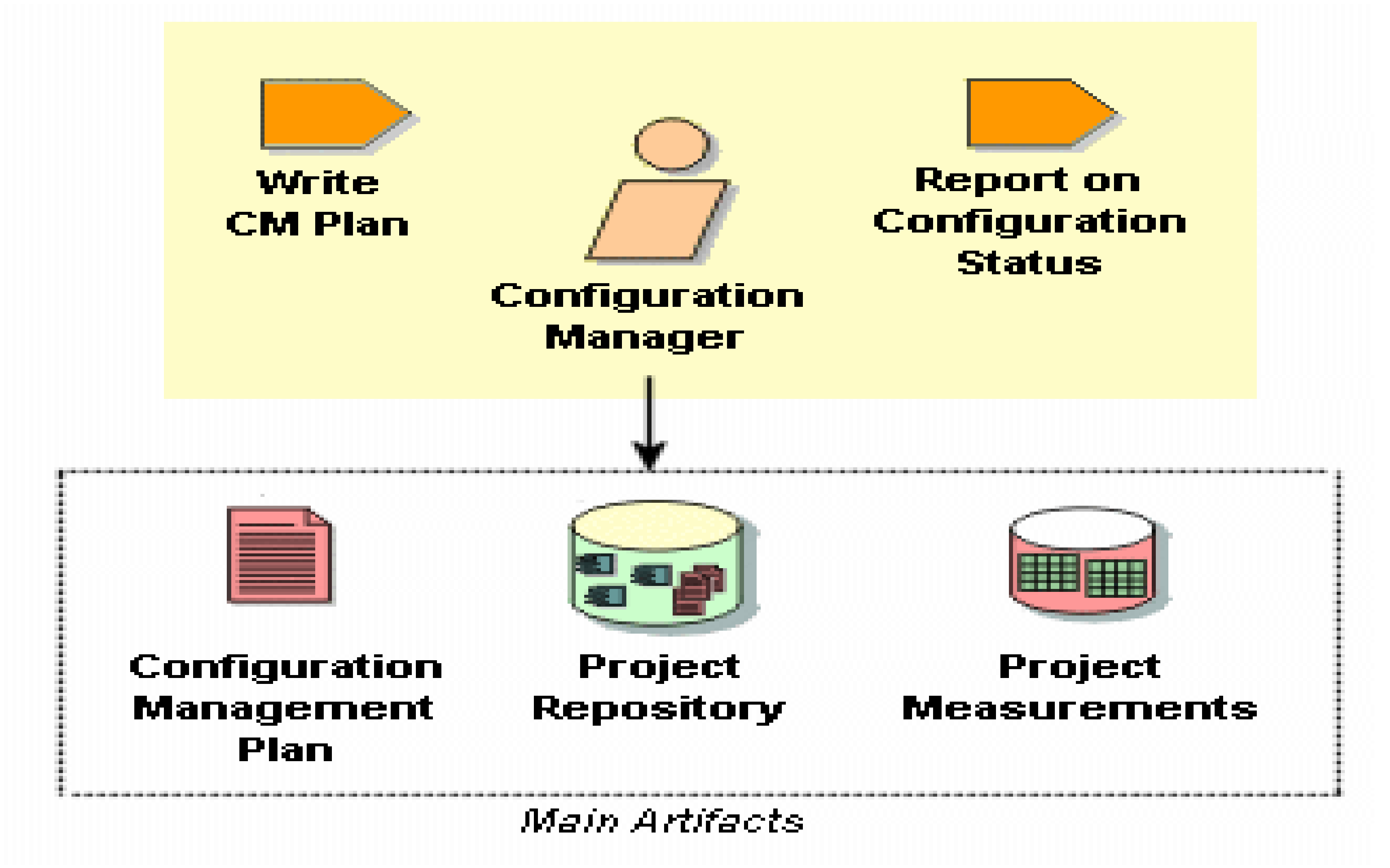


Software Configuration Manager

- The **configuration manager** provides the overall Configuration Management (CM) infrastructure and environment to the product development team. The CM role supports the product development activity so that developers and integrators have appropriate workspaces to build and test their work, and so that all artifacts are available for inclusion in the deployment unit as required. The configuration manager also has to ensure that the CM environment facilitates product review, and change and defect tracking activities. The configuration manager is also responsible for writing the CM Plan and reporting progress statistics based on change requests.

SUPPORT
DISPOSE
PRODUCE
Configuration Management Lifecycle
DESIGN
VALIDATE

Software Configuration Manager



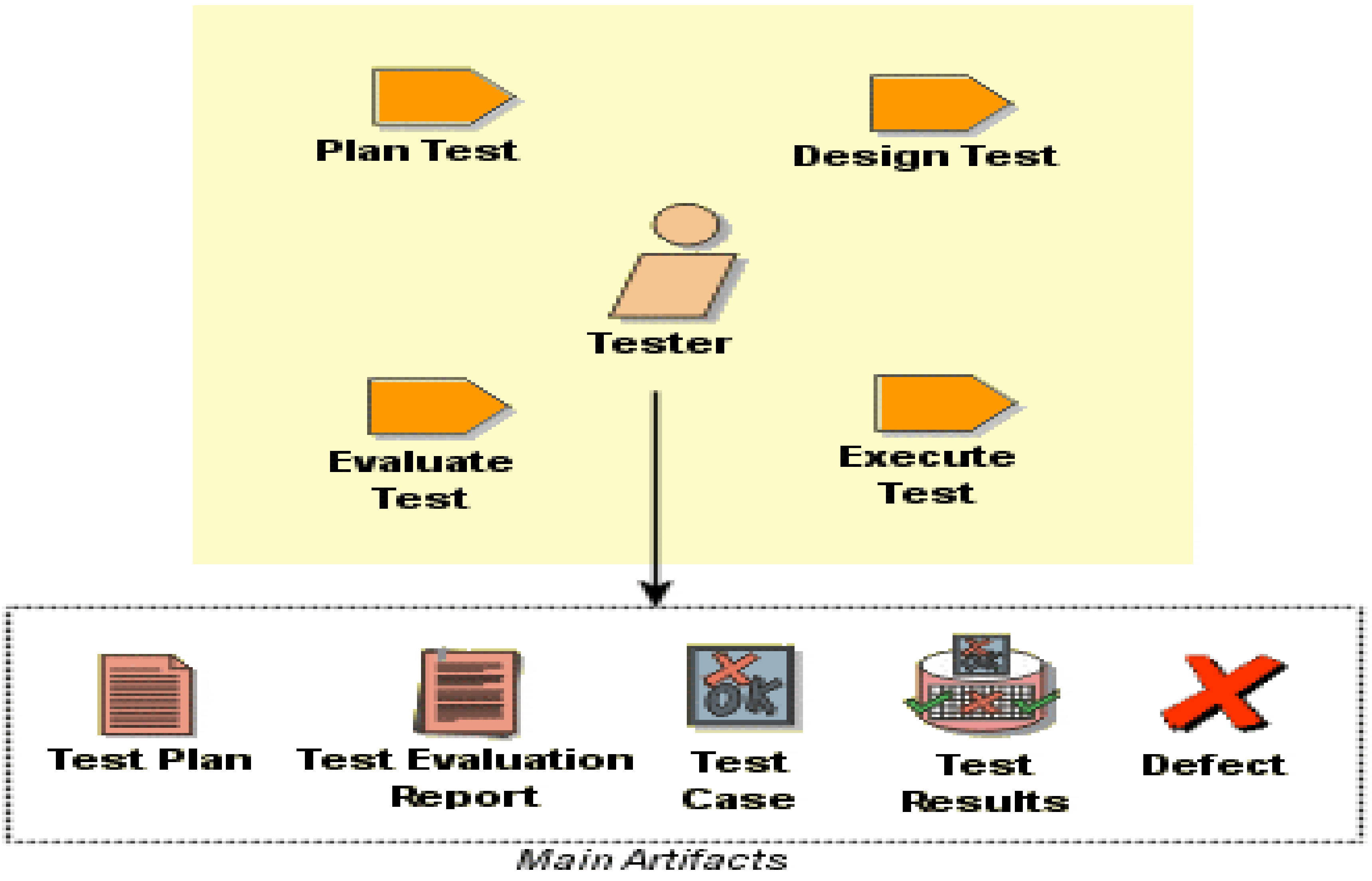
Software Tester

- A **software tester** is an individual that tests software for bugs, errors, defects or any problem that can affect the performance of computer software or an application.
- Software testers are part of a software development team and perform functional and non-functional testing of software using manual and automated software testing techniques.
- S/he is responsible for
  - Identifying the most appropriate implementation approach for a given test
  - Implementing individual tests
  - Setting up and executing the tests
  - Logging outcomes and verifying test execution
  - Analyzing and recovering from execution errors

Some of the techniques software testers must have experience with include:

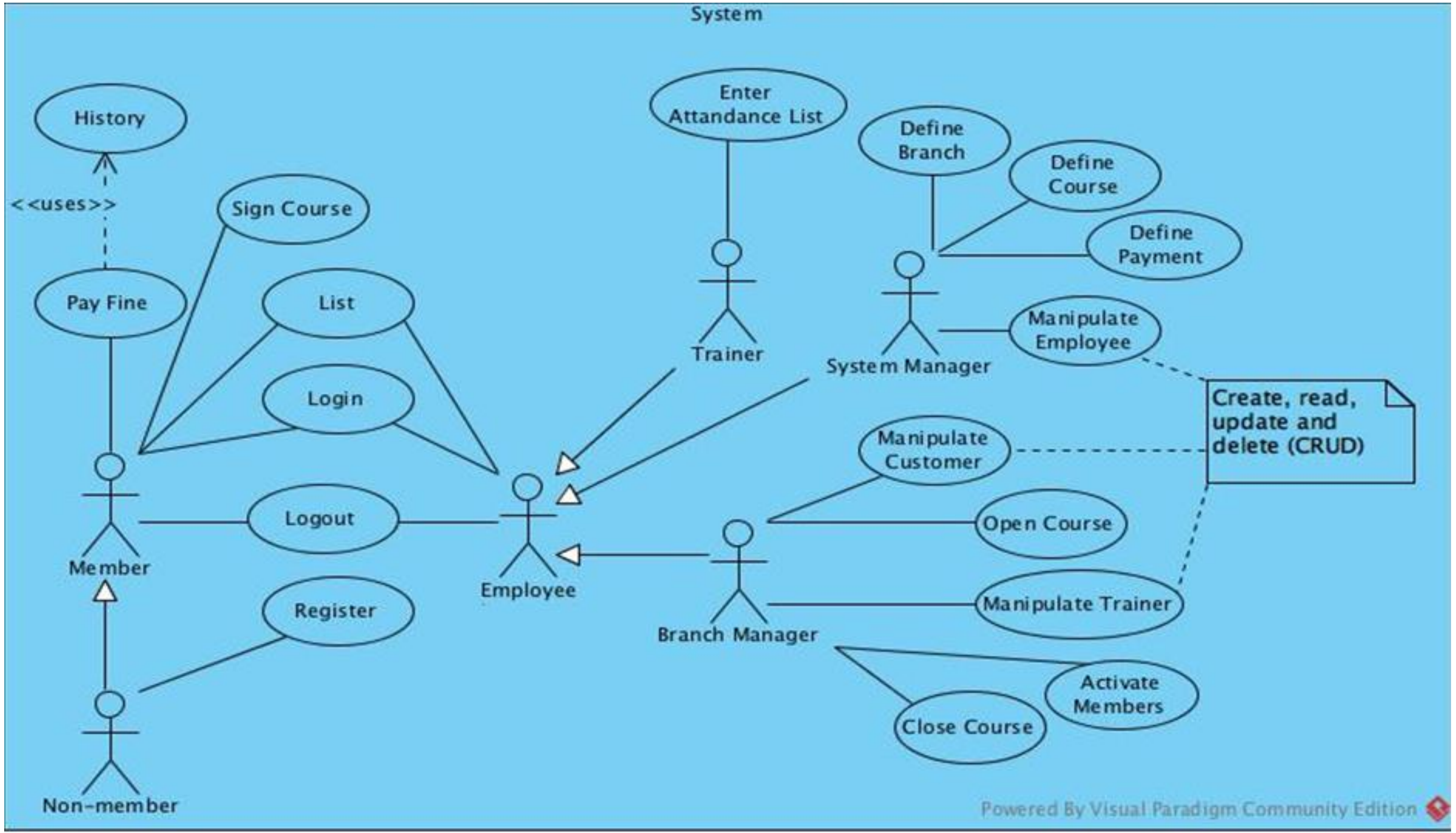
- Unit Testing
- System Testing
- Black box Testing
- Load Testing
- User Acceptance Testing
- Scalability Testing

Software Tester



Sports Center Membership System (SCMS) – Requirements

UML Use Case Diagram for SCMS



See you next week…