S1 Informatic Engineering

Advanced Software Engineering

Web App. Process and Architecture

By:
Egia Rosi Subhiyakto, M.Kom, M.CS
Informatic Engineering Department
egia@dsn.dinus.ac.id
+6285640392988

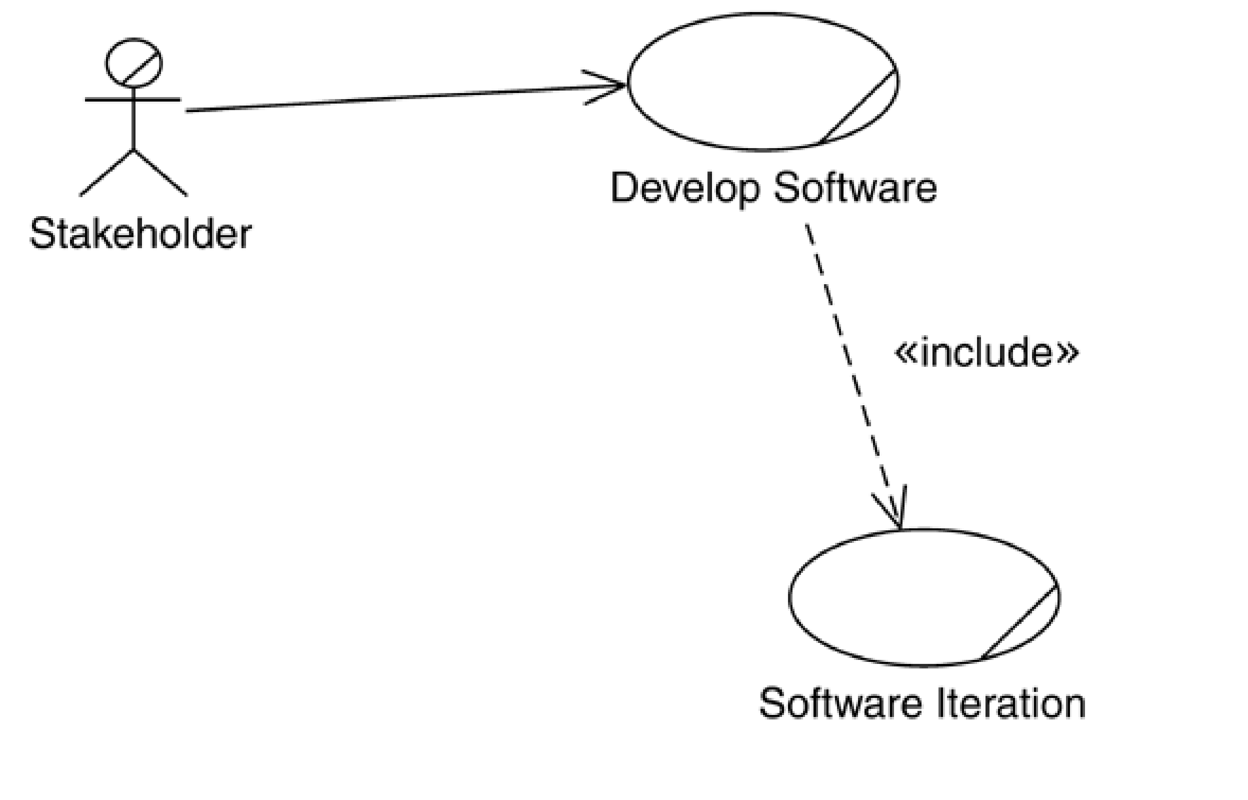
SYLLABUS

8. Web App. Process and Architecture
9. WebE Design (1)
10. WebE Design (2)
11. Real Time Software
12. Testing Web App.
13. Present Tugas Besar
14. Present Tugas Besar

Agenda

- The process
  - Express the process in terms of UML model
- Defining the architecture
  - The activities
  - Web app architectural pattern
  - Presentation tier pattern

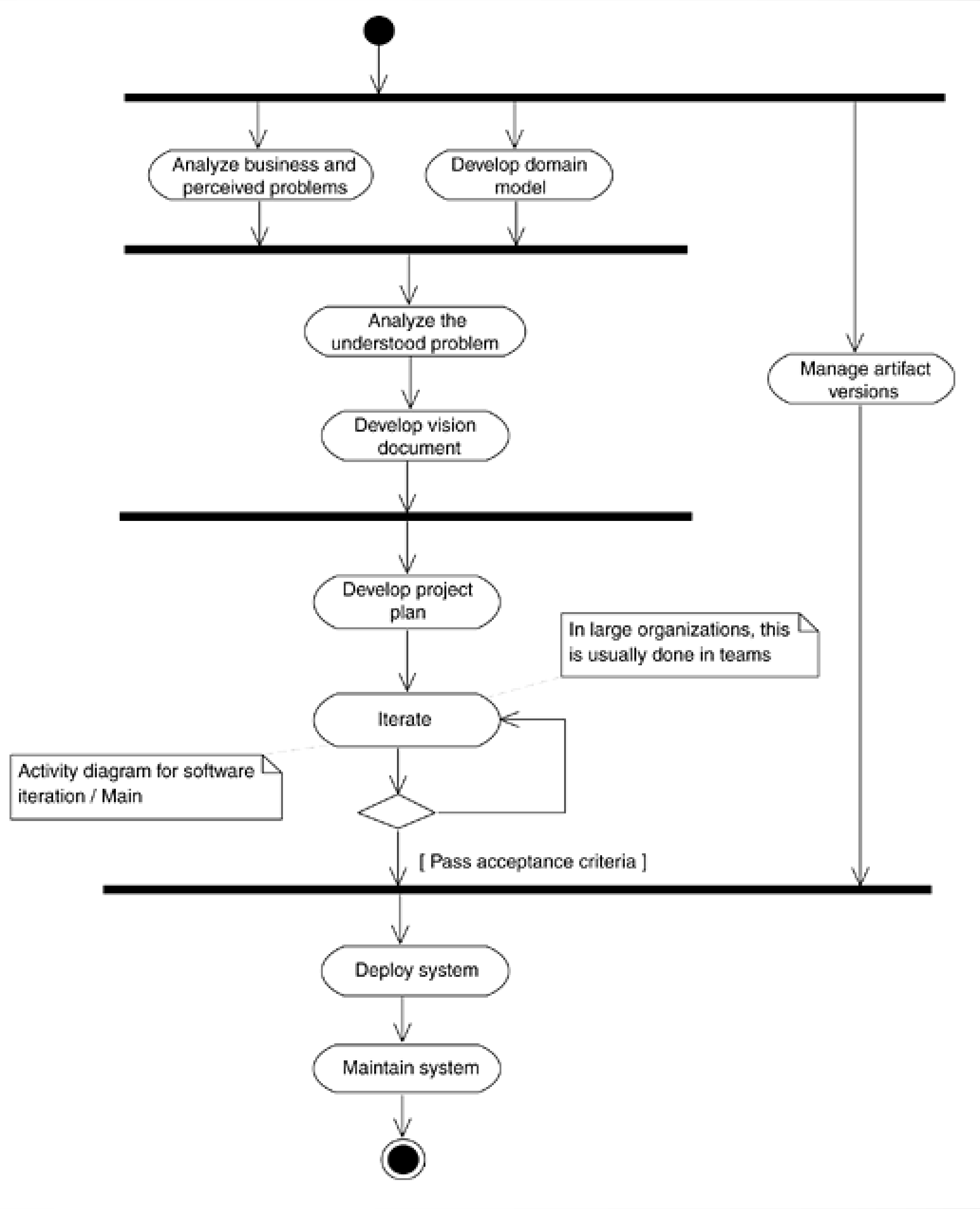
Software Development for Web App



Software Development for Web App (2)

- The business actor Stakeholder represents all individuals outside of the project who have a vested interest in the project.
  - They don't build the software; nor are they responsible for any artifacts of the process.
  - They contribute through interactions with the workers in the process.
  - The Stakeholders are the customers, company executives, investors, and users: anyone with an active interest in the evolution and the delivery of the system.
- A Stakeholder interacts with the "business" through the business use case Develop Software, which is the main use case of the process.
- A critical part of this use case is the included use case Software Iteration.
  - Software Iteration describes the bulk of the work that we would normally consider as part of the development process.

Activities of The Develop Software Use Case


Analyze business and perceived problems
Develop domain model
Analyze the understood problem
Manage artifact versions
Develop vision document
Develop project plan
In large organizations, this is usually done in teams
Iterate
Activity diagram for software iteration / Main
[Pass acceptance criteria]
Deploy system
Maintain system

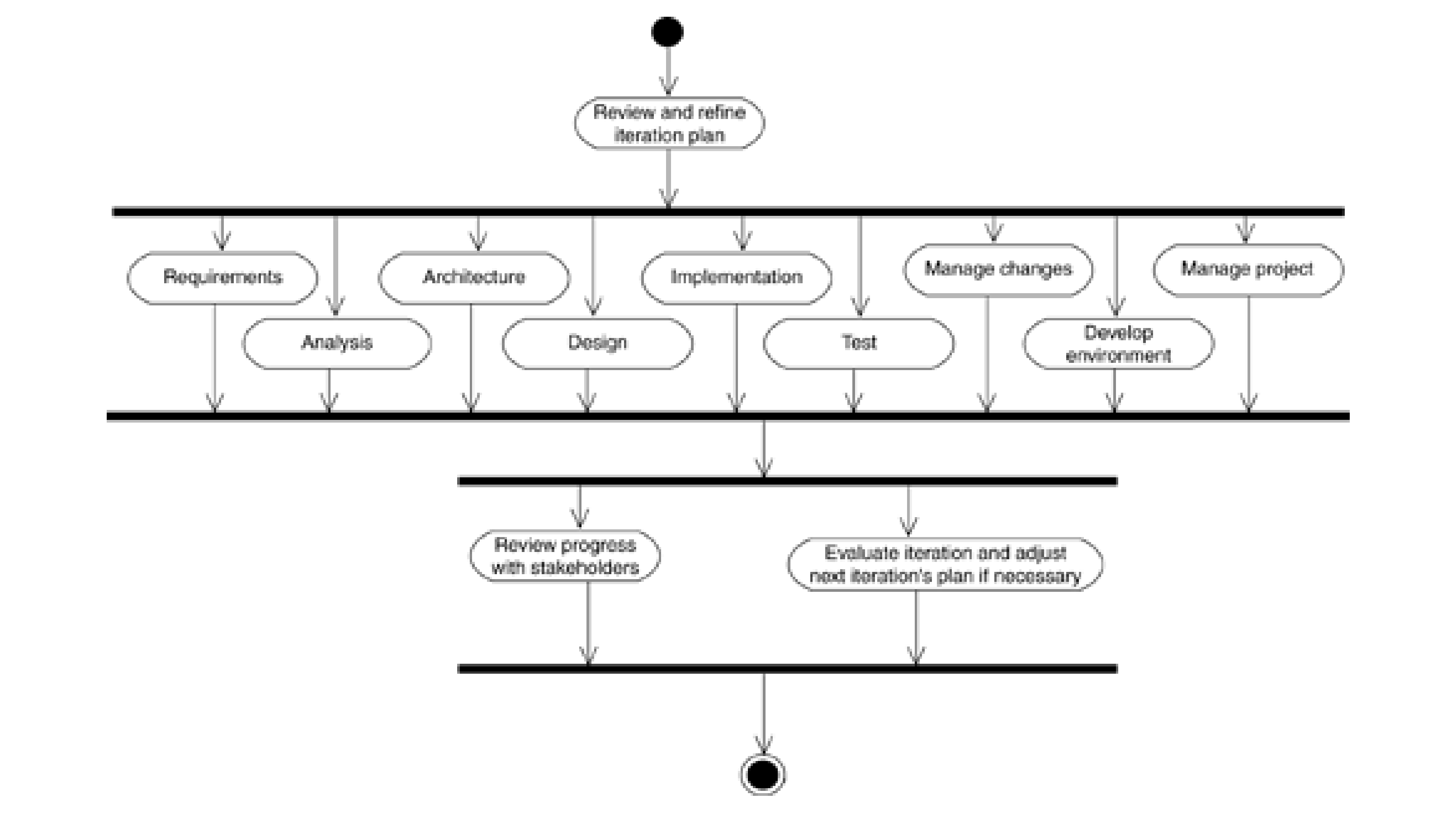
Activities of The Develop Software Use Case (2)

- Manage artifact versions.
  - Done concurrently with every other activity
  - It is the activity of the change management process and the use of a version control system.
- Analyze business and perceived problems.
  - Take an objective look at the state of the business.
  - Try not to be influenced by the perceived problems presented by the stakeholders.
- Develop domain model.
  - Use software development tools, such as UML, requirements gathering, and document management applications, to construct a model of the domain and the business.
  - The UML extension for business modeling is the most appropriate way to express the major entities and processes of the business.
  - The most important artifact: the glossary, which defines the key terms and concepts of the context that the system is working in.

Activities of The Develop Software Use Case (3)

- Analyze the understood problem.
  - Focus on the real problem, or at least the problem that the stakeholders are willing to let you work on
- Develop vision document.
  - The vision document expresses the scope and the purpose of the entire software project.
- Develop project plan.
  - The project plan outlines the activities of the entire software development effort, defining the major milestones and referencing appropriate standards documents, including the change and configuration management plan.
- Deploy system.
  - Deliver and install the system.
- Maintain system.
  - System maintenance is essentially a miniversion of the process that developed it.

Activities of The Software Iteration Use Case


Review and refine iteration plan
Requirements
Analysis
Architecture
Design
Implementation
Test
Manage changes
Develop environment
Manage project
Review progress with stakeholders
Evaluate iteration and adjust next iteration's plan if necessary

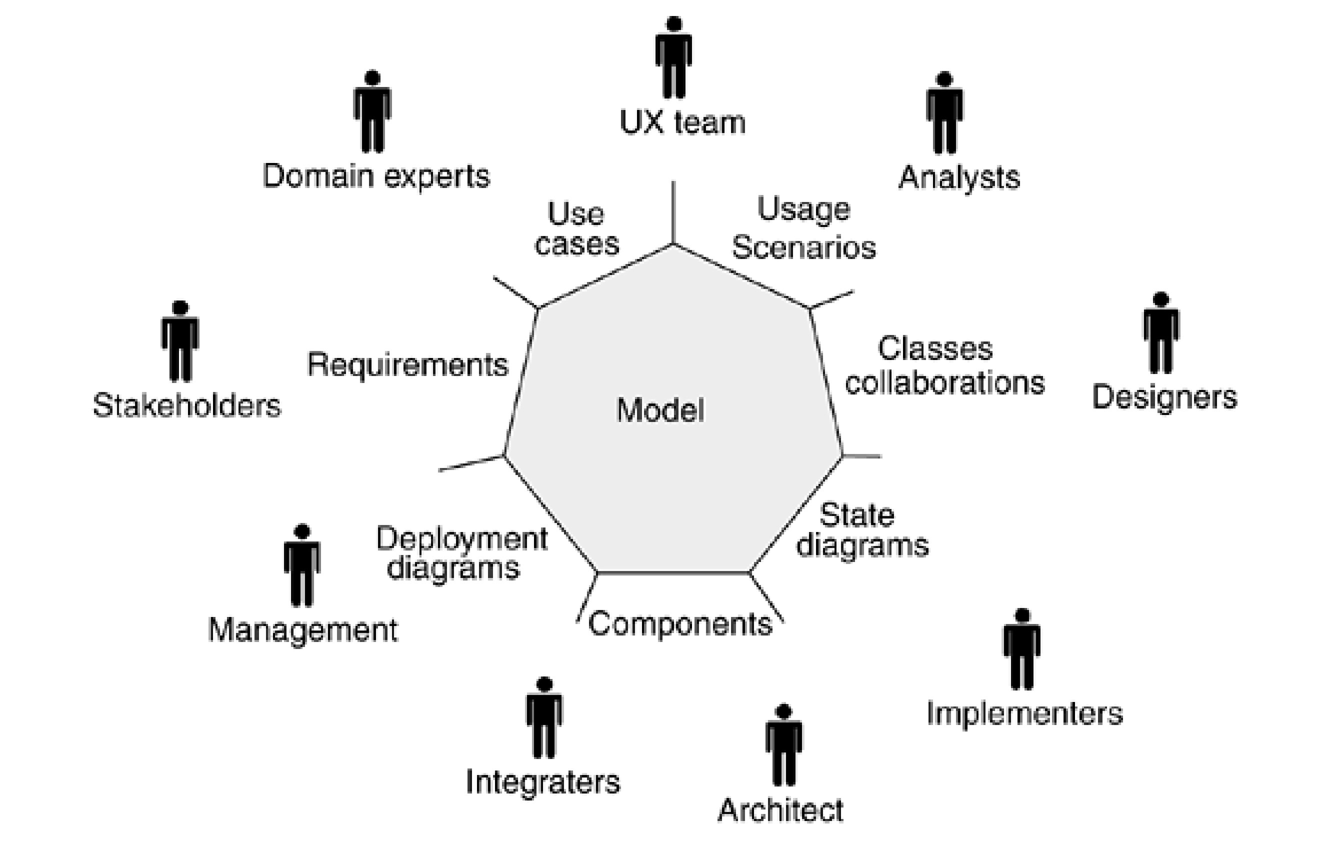
Activities of The Software Iteration Use Case (2)

- Review and refine the iteration plan.
  - Before the iteration, the team takes a look at the iteration plan.
  - The project manager makes resource allocations, and the team makes its deliverable commitments.
- Review progress with stakeholders.
  - Often the stakeholders are very interested in the progress of the project.
- Evaluate iteration and adjust next iteration's plan.
  - Have the deliverables been made with the expected effort?
  - Is the level of quality what was expected by the team?
  - Did it meet your personal expectations?
  - These are all important measures that need to be reflected in the next iteration's plan.

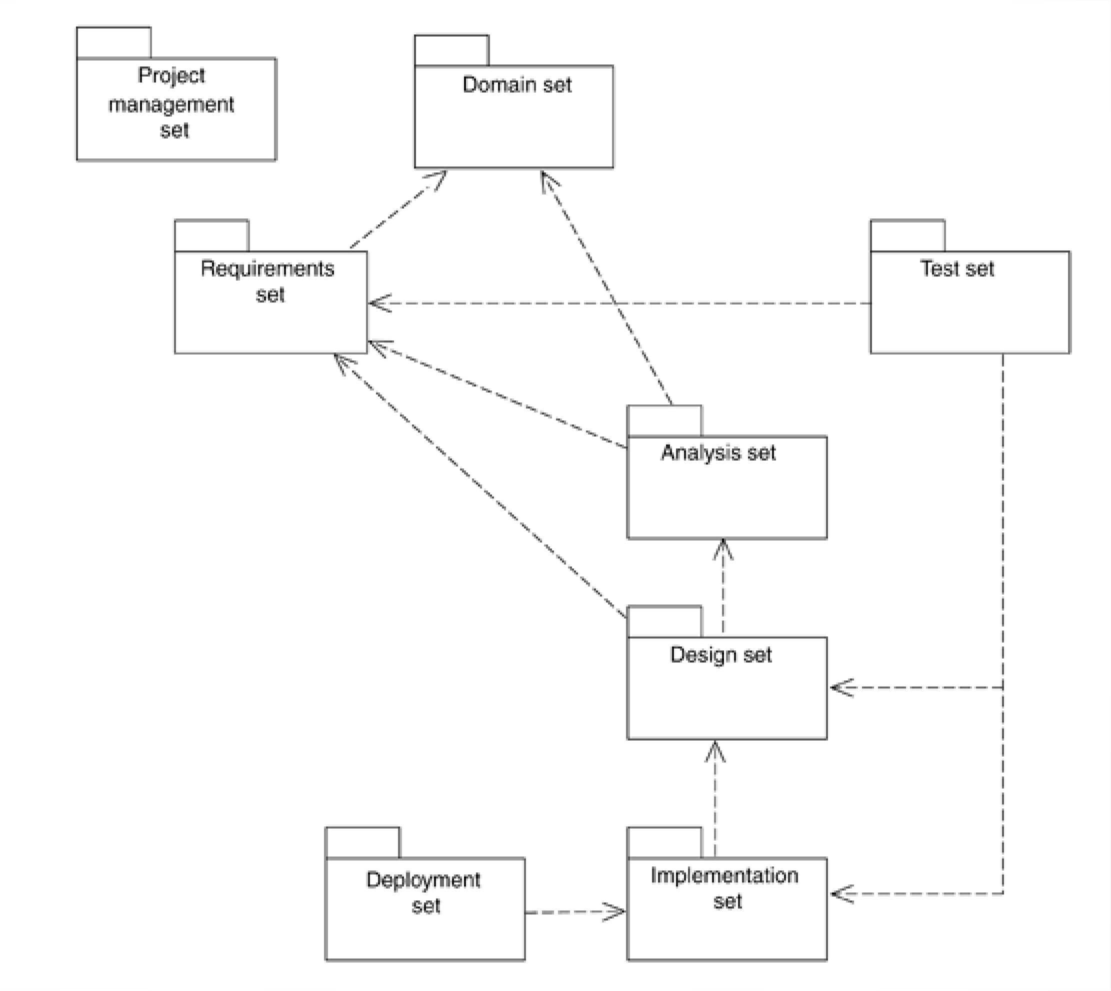
Artifact of The Development Process

- In the Web application development process, the recommended models are:
  - Domain model
  - Use case model
  - Analysis/design model
  - Implementation model
  - Process model
  - Security model
  - User experience, or user interface, model

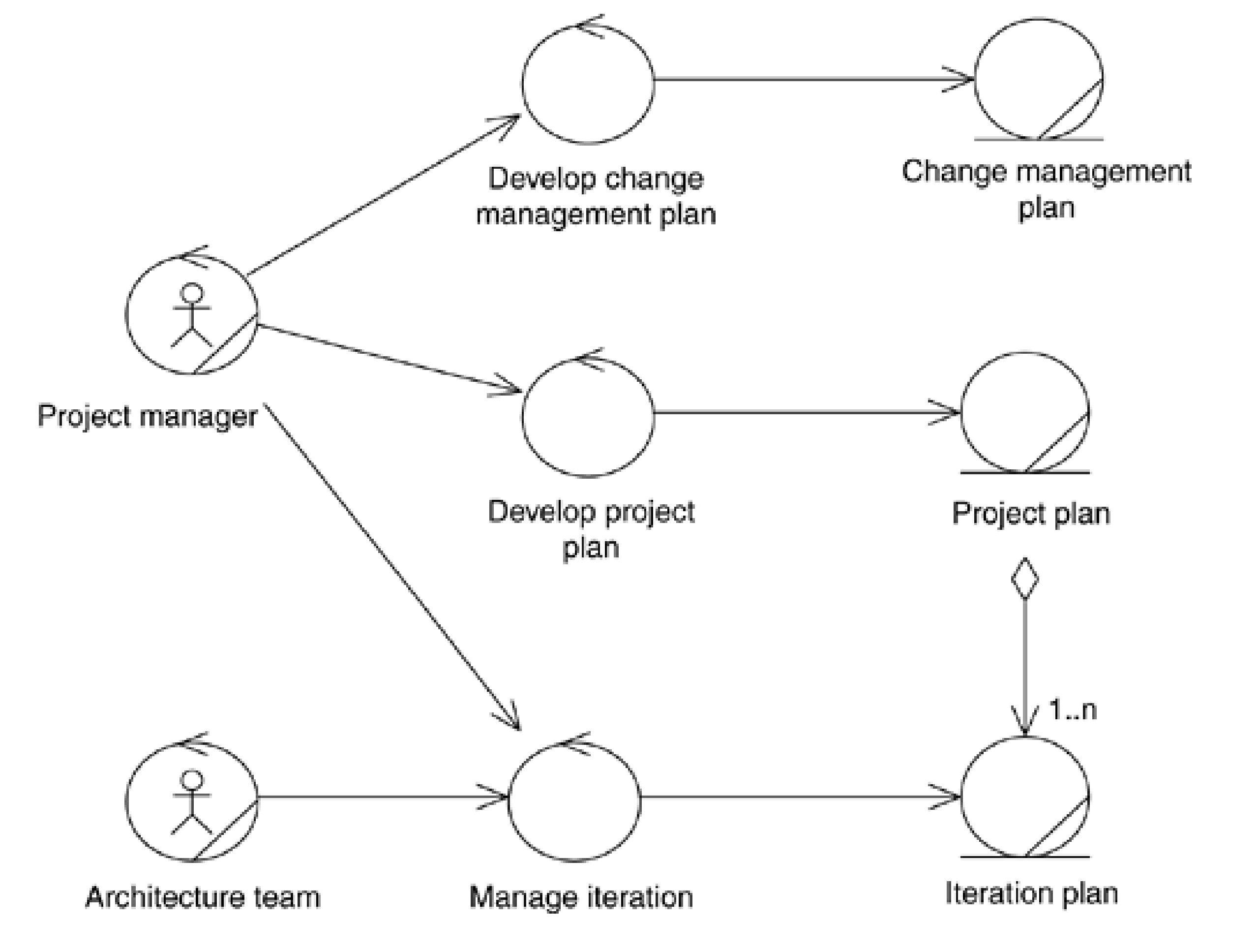
Artifact of The Development Process (2)



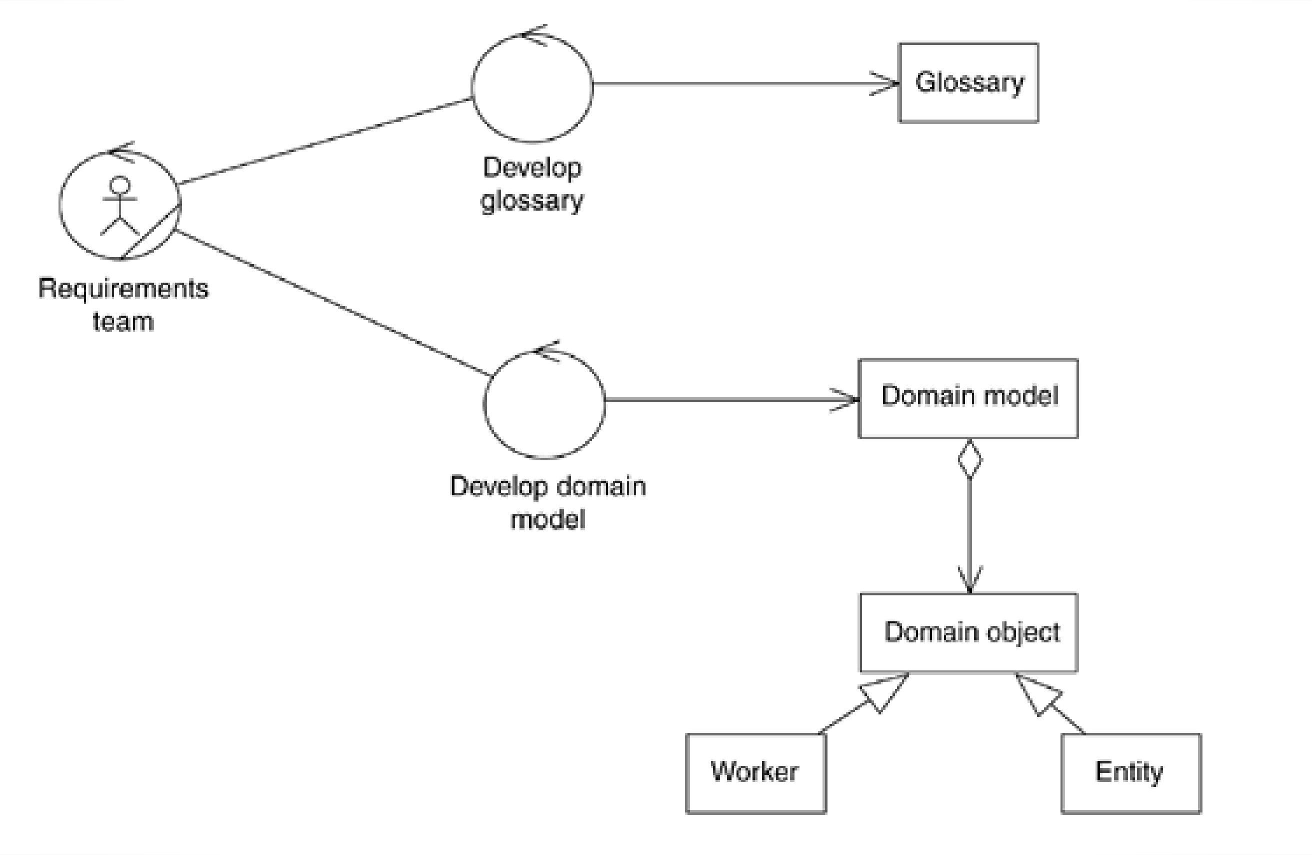
Dependencies and traceabilities in artifact sets



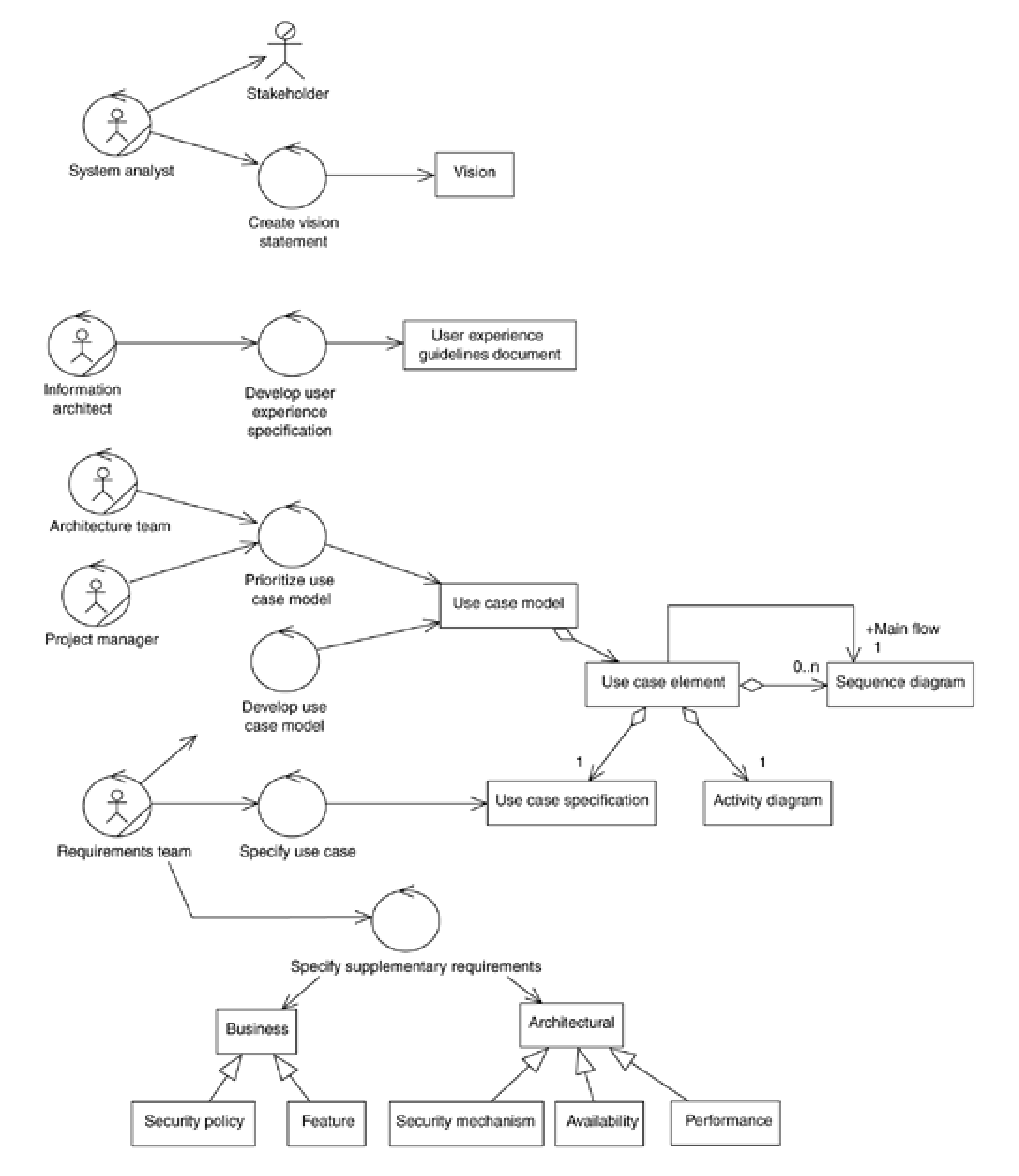
Project Management Set



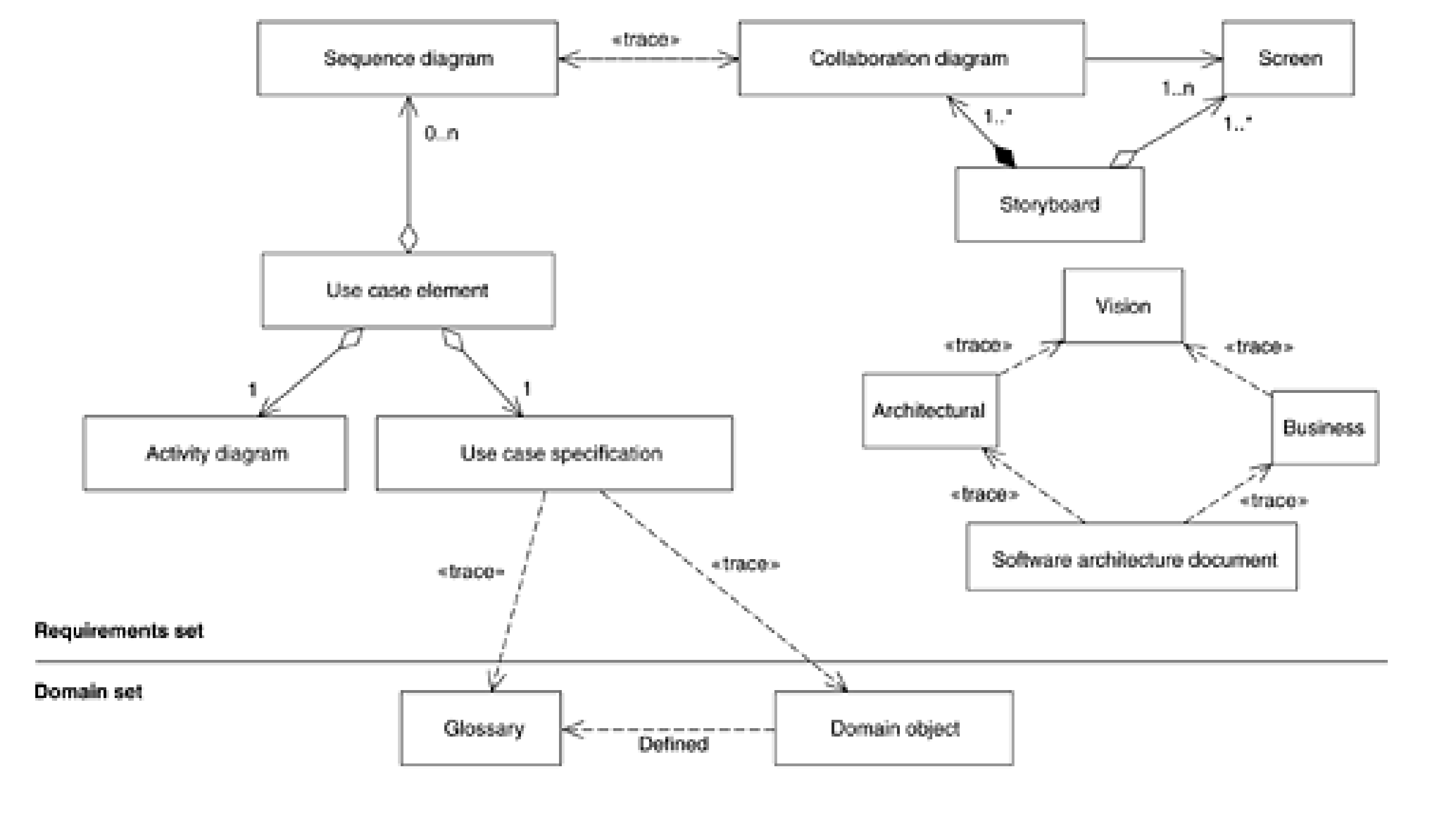
Domain set


Requirements team
Develop glossary
Glossary
Develop domain model
Domain model
Domain object
Worker
Entity

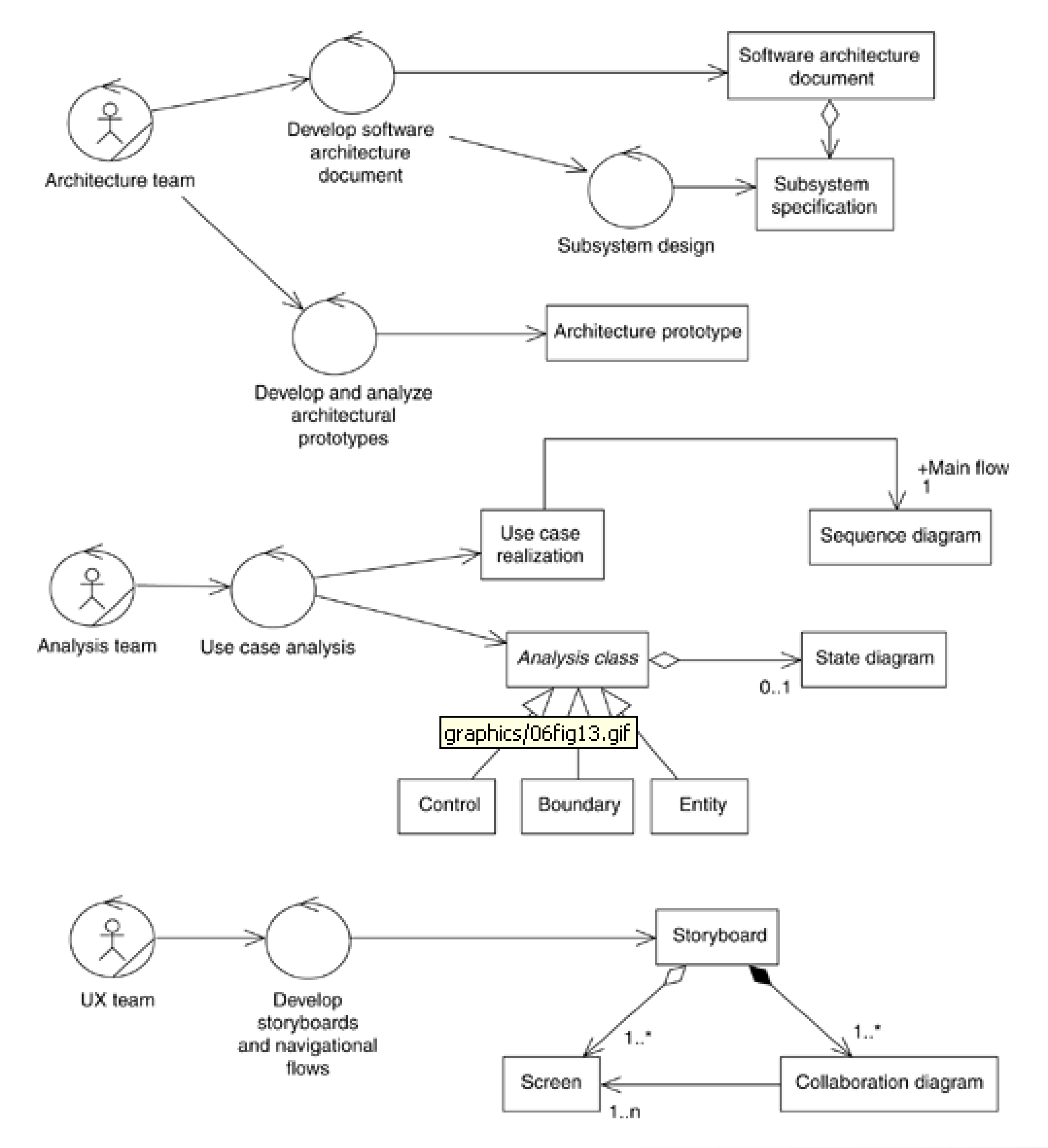
Requirements set



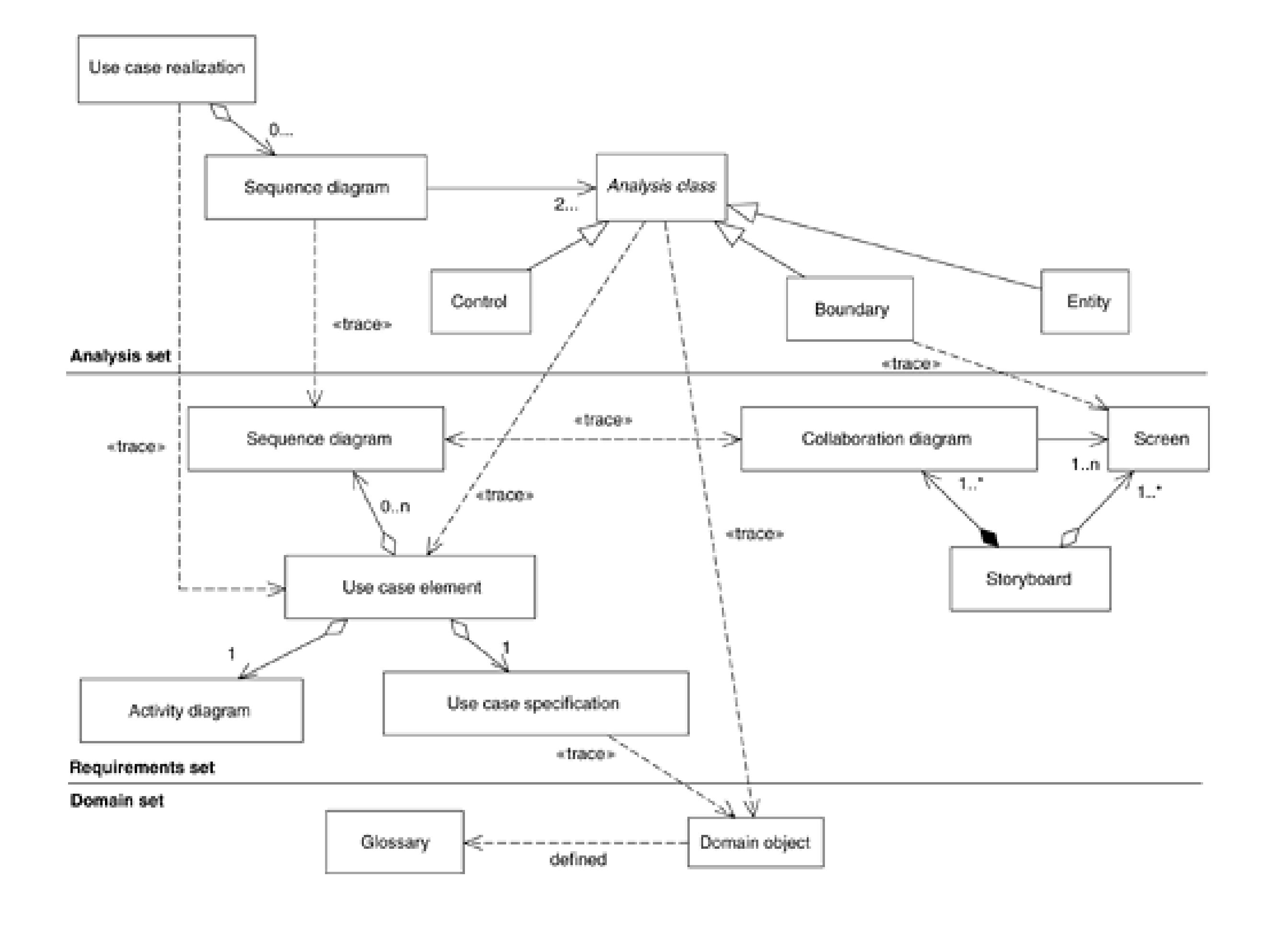
Associations and traceabilities in the requirements set



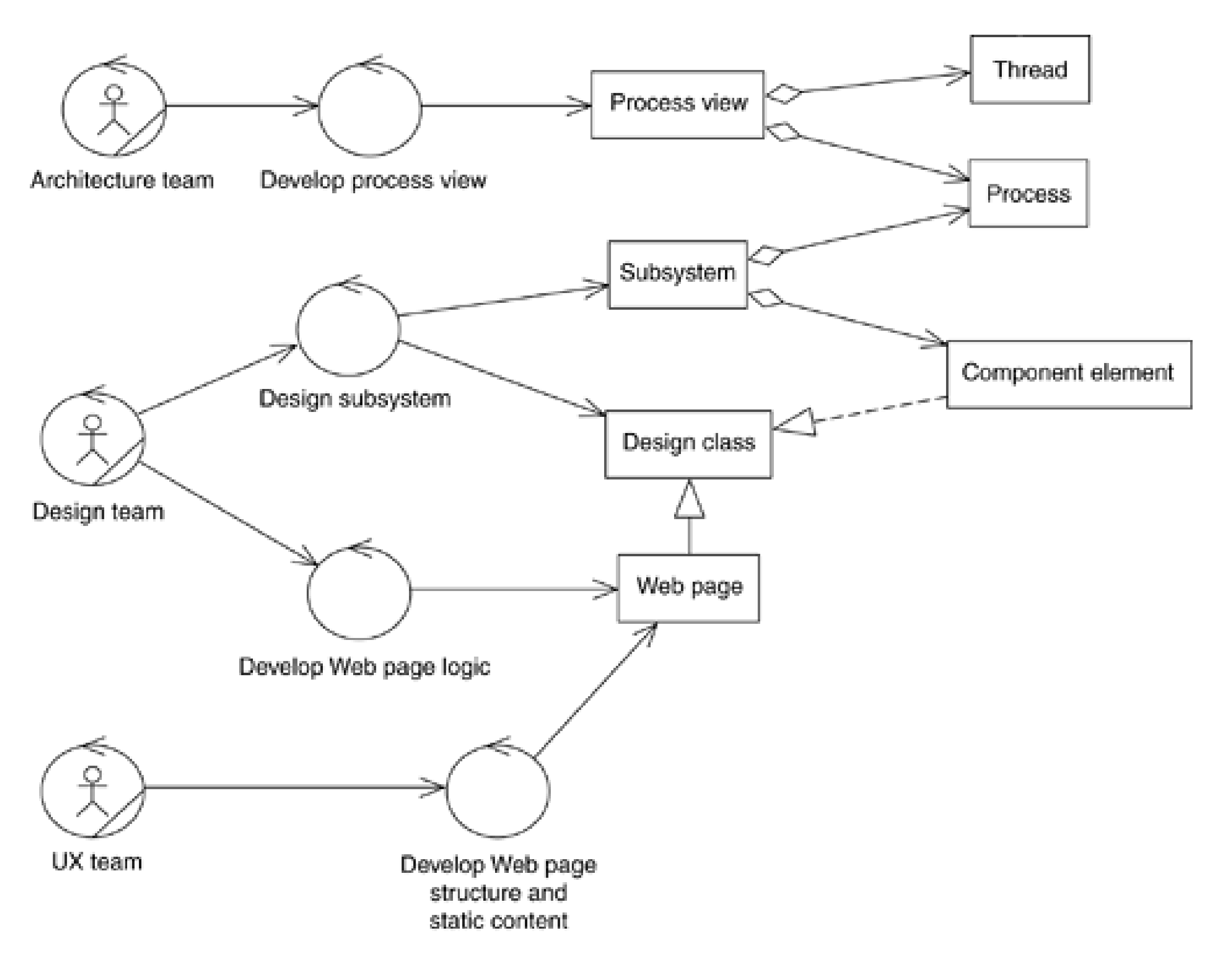
Analysis set



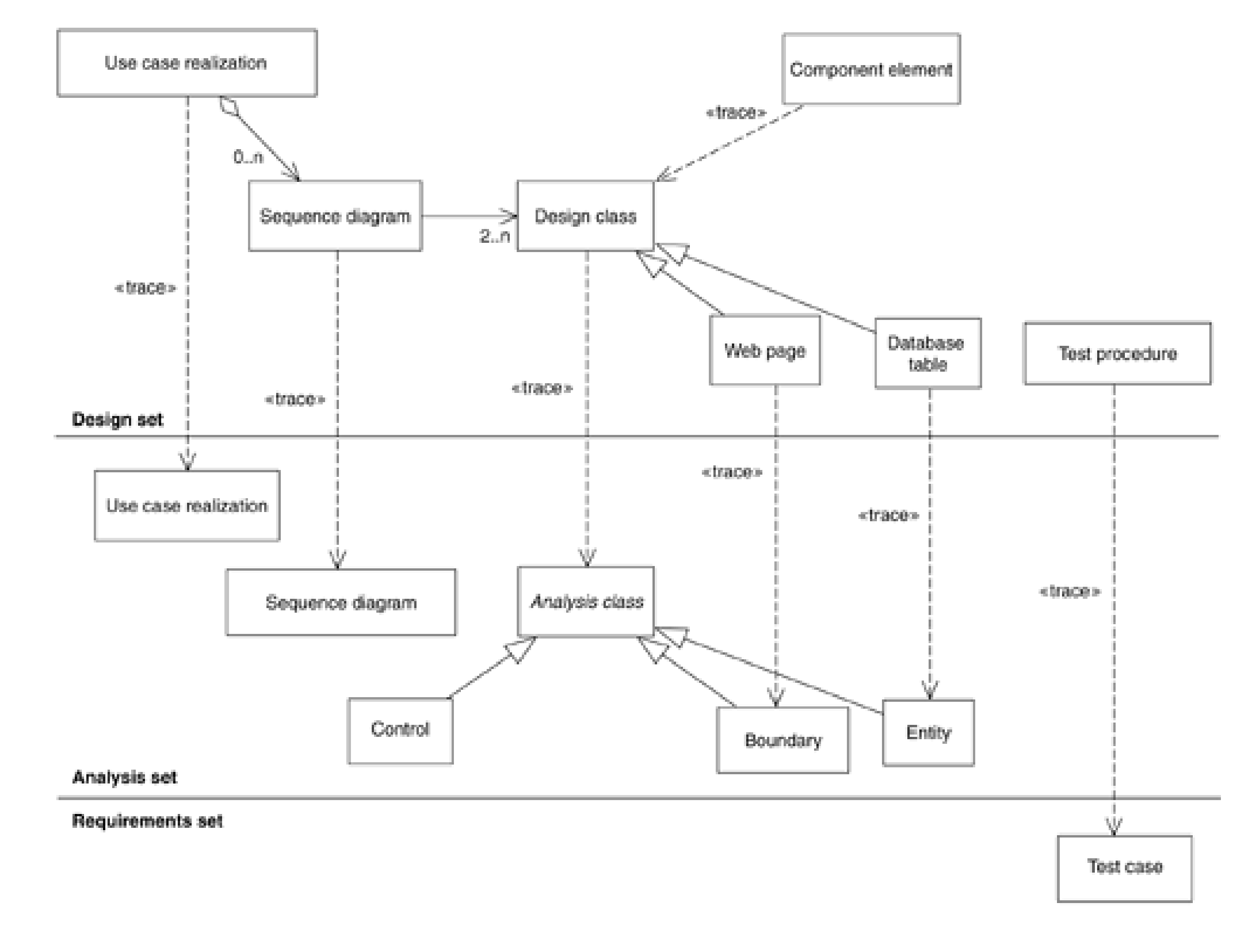
Associations and traceabilities in the analysis set



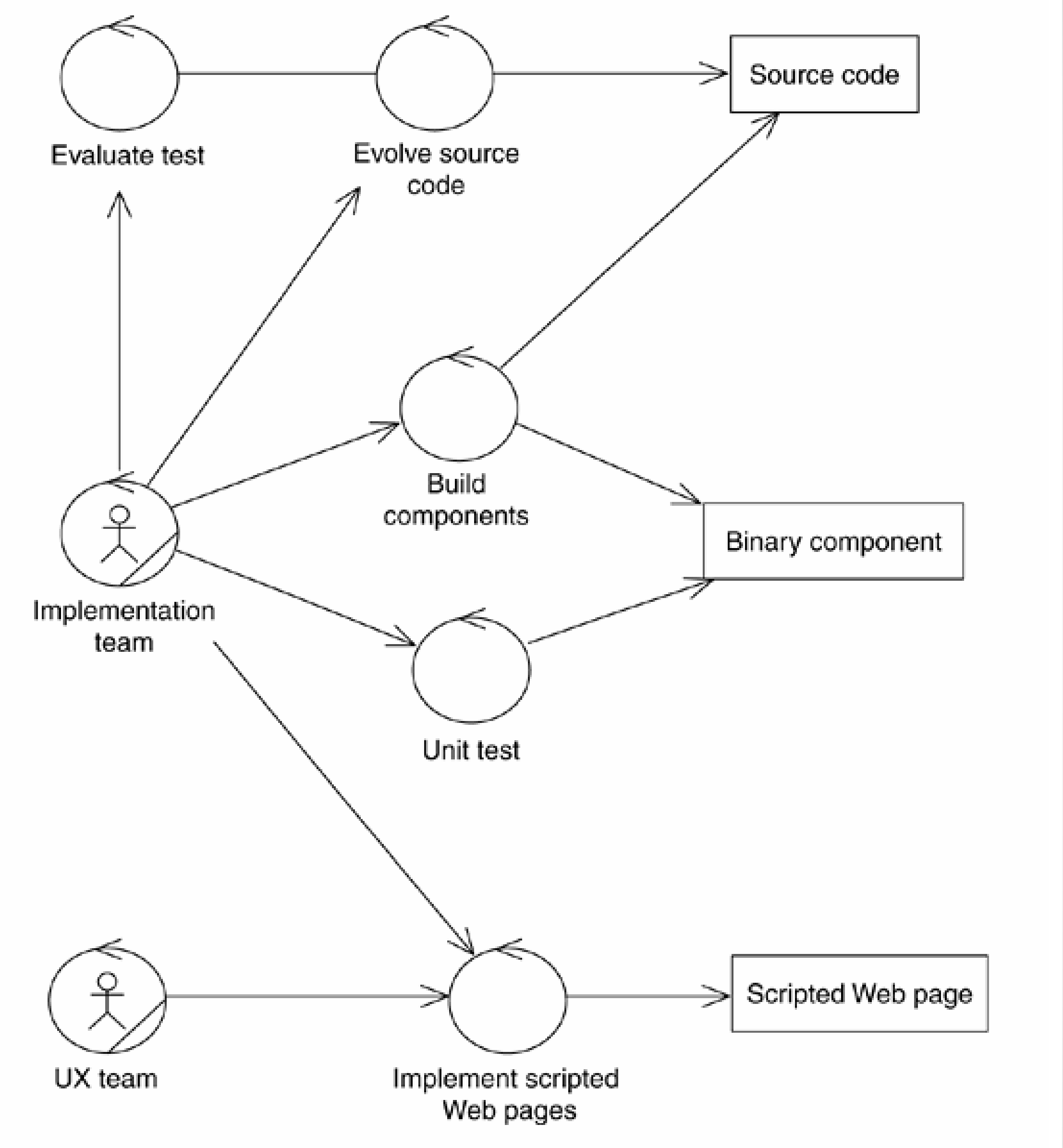
Design set


Architecture team
Develop process view
Process view
Thread
Process
Subsystem
Component element
Design subsystem
Design class
Design team
Web page
Develop Web page logic
UX team
Develop Web page structure and static content

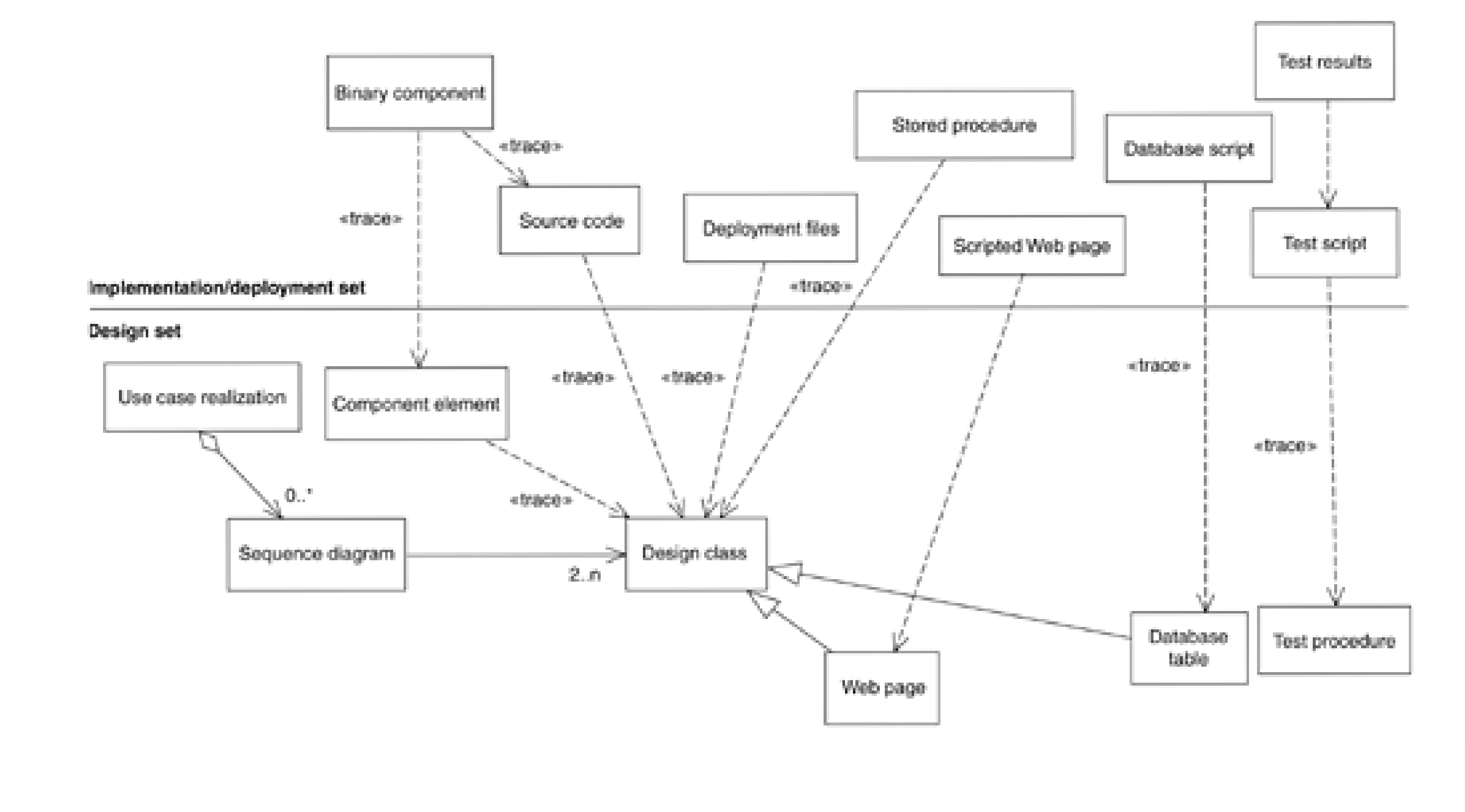
Associations and traceabilities in the design set



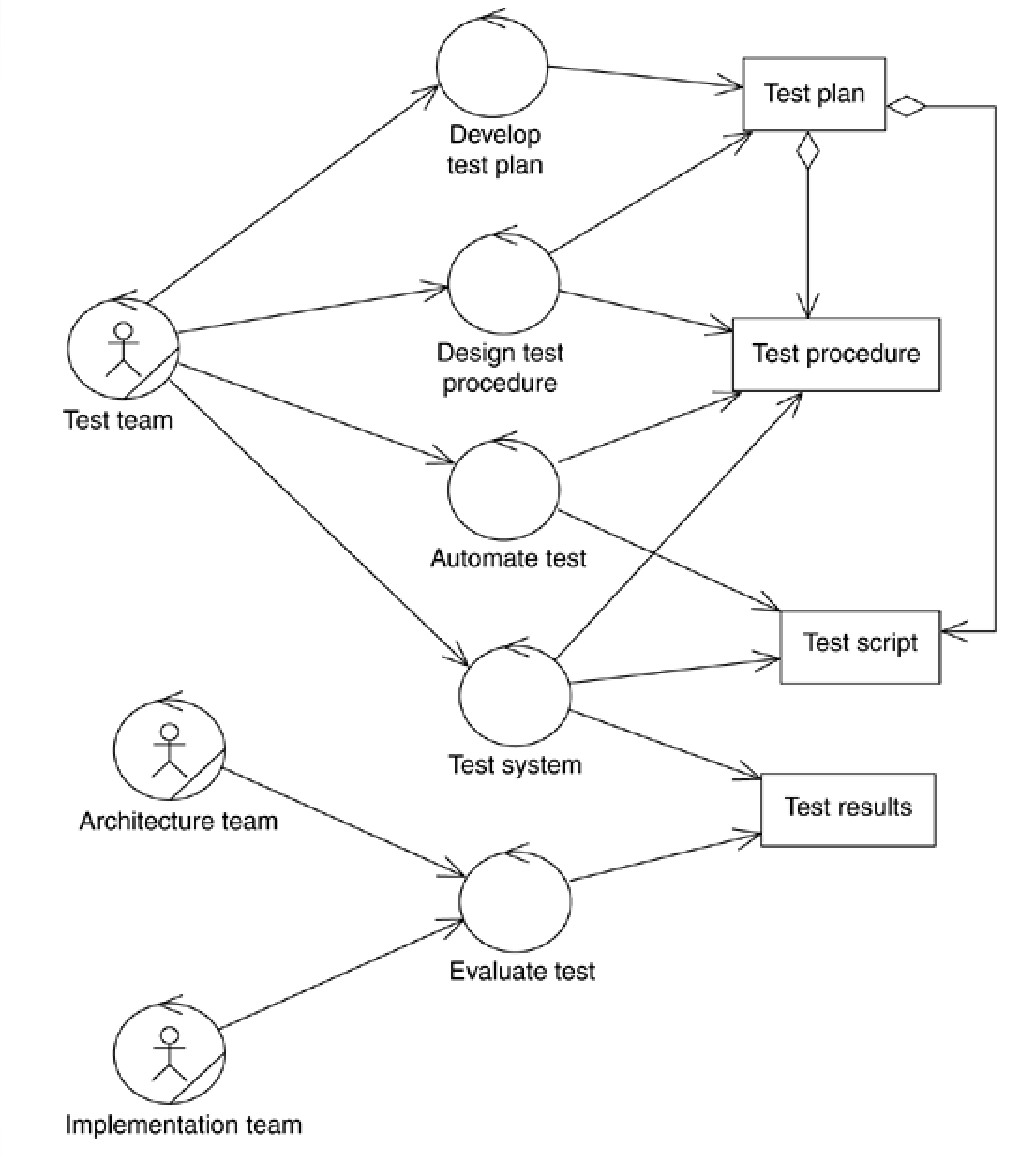
Implementation set



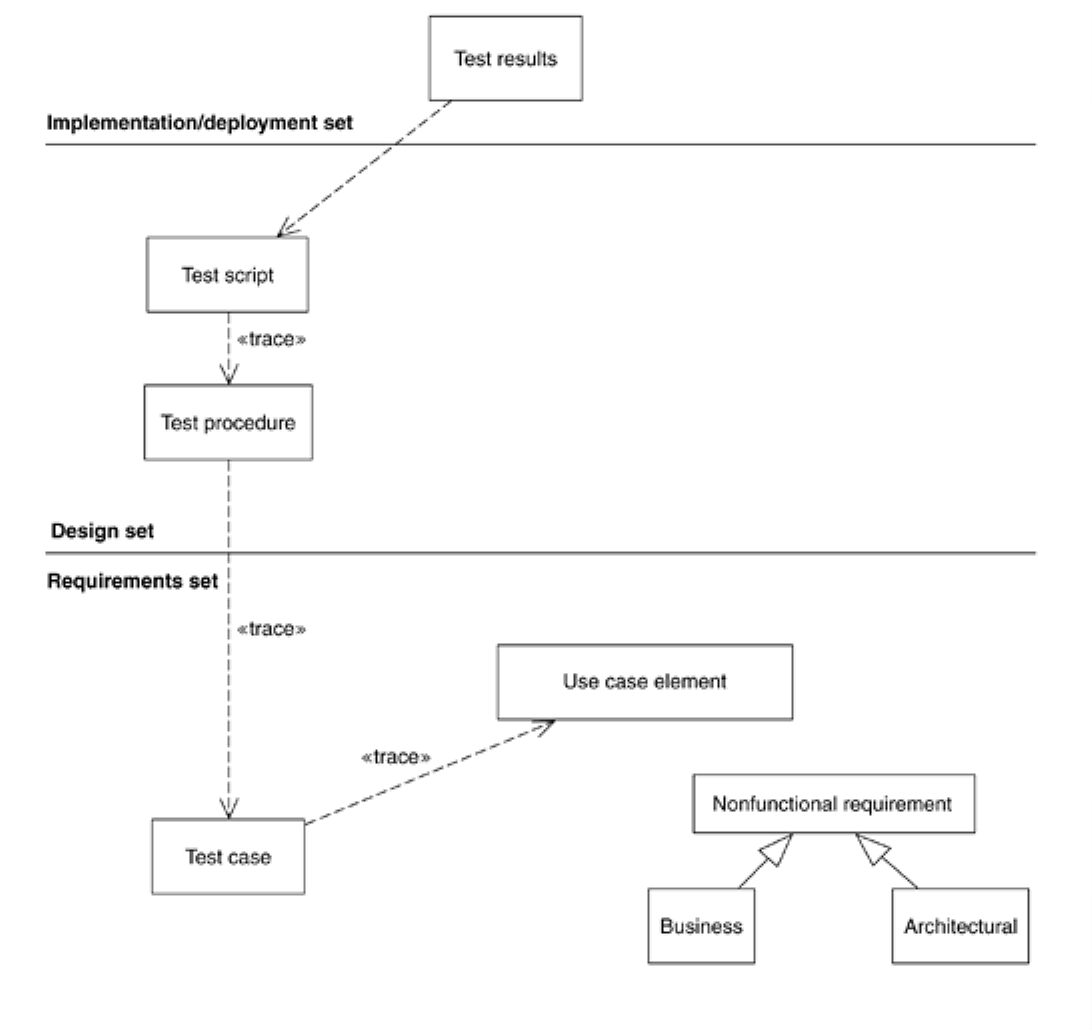
Associations and traceabilities in the implementation set



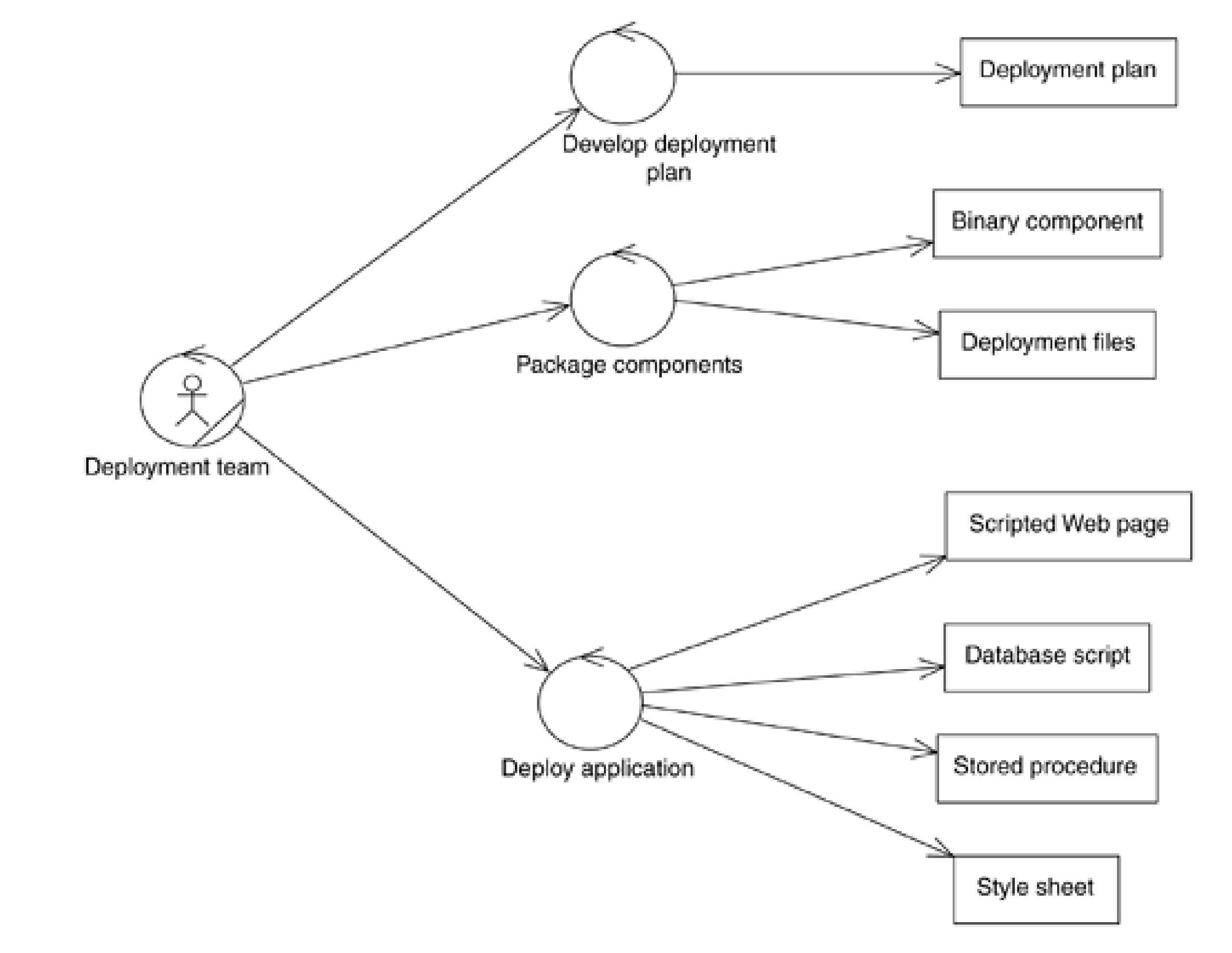
Test set


Develop test plan
Test plan
Design test procedure
Test procedure
Test team
Automate test
Test script
Test system
Test results
Architecture team
Evaluate test
Implementation team

Associations and traceabilities in the test set



Deployment set



Defining The Architecture

- Architecture activities
- Web app architectures pattern:
  - Façade
  - Page Composition
  - Templated Page
- Presentation tier pattern:
  - Thin Web Client
  - Thick Web Client
  - Web Delivery

Architecture Activities

- Examine and prioritize use cases and look for significant architectural requirements
- Define and document a candidate architecture and prepare and evaluate prototypes
- Define a reuse strategy

Web Application Architectural Patterns

- Façade
  - The dynamic information in any given Web page may have to be constructed from a collection of business objects and controllers.
  - Façade classes pair up with dynamic Web pages. Each Web page has a specifically designed façade class that acts to consolidate all the business object orchestration and to provide a clear, easy-to-use interface for the Web page script writer to use.
- Page composition.
  - Each conceptual Web page in the system is assembled at runtime from a set of independent smaller page fragments, which are often reused across pages in the system.
  - For example, many Internet retail applications provide a quick way to enter product search criteria on every conceptual Web page.
- Templated page.
  - This pattern defines a one-page template that all outgoing Web pages go through on their way to the client.
  - Similar to the page composition pattern, the templated page pattern provides additional structure with formally defined templates and screens (conceptual pages).

Patterns for The Presentation Tier

- The thin Web client is used mostly for Internet-based applications, in which there is little control of the client's configuration.
  - The client requires only a standard forms-capable Web browser.
  - All the business logic is executed on the server.
- The thick Web client pattern is used when an architecturally significant amount of business logic is executed on the client machine.
  - Typically, the client uses dynamic HTML, Java applets, or ActiveX controls to execute business logic.
- The Web delivery pattern is used when the Web browser acts principally as a delivery and container device for a distributed object system.
  - In addition to HTTP for client and server communication, other protocols, such as IIOP and DCOM, may be used to support a distributed object system.

Thin Web Client

Requirements Viewpoint

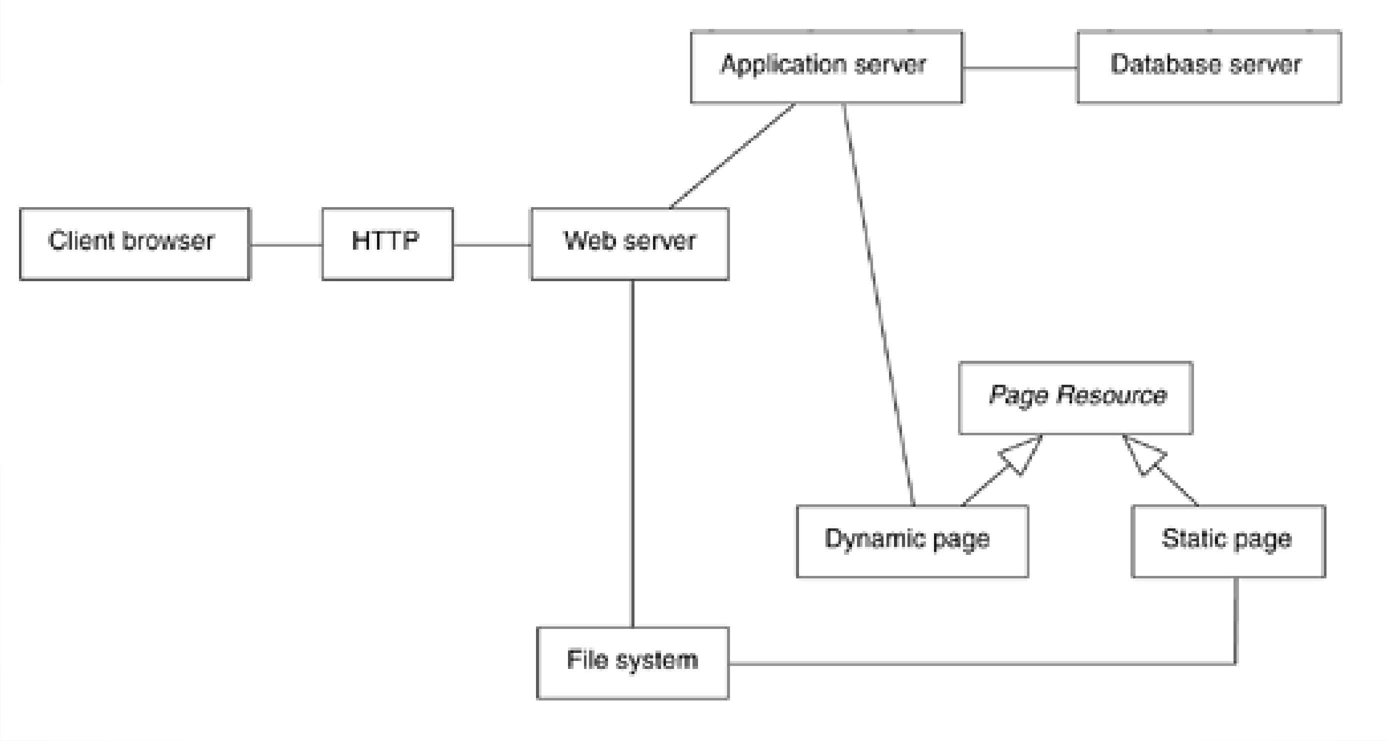
- The principal requirements: the ability to deliver the system to as many types of user systems as possible
- Thin Web client application requires only minimal capabilities on the client
- Additional restrictive requirements: the ability for the browser to accept and to return cookies; the ability to operate in low-bandwidth environments; etc

Thin Web Client (2)

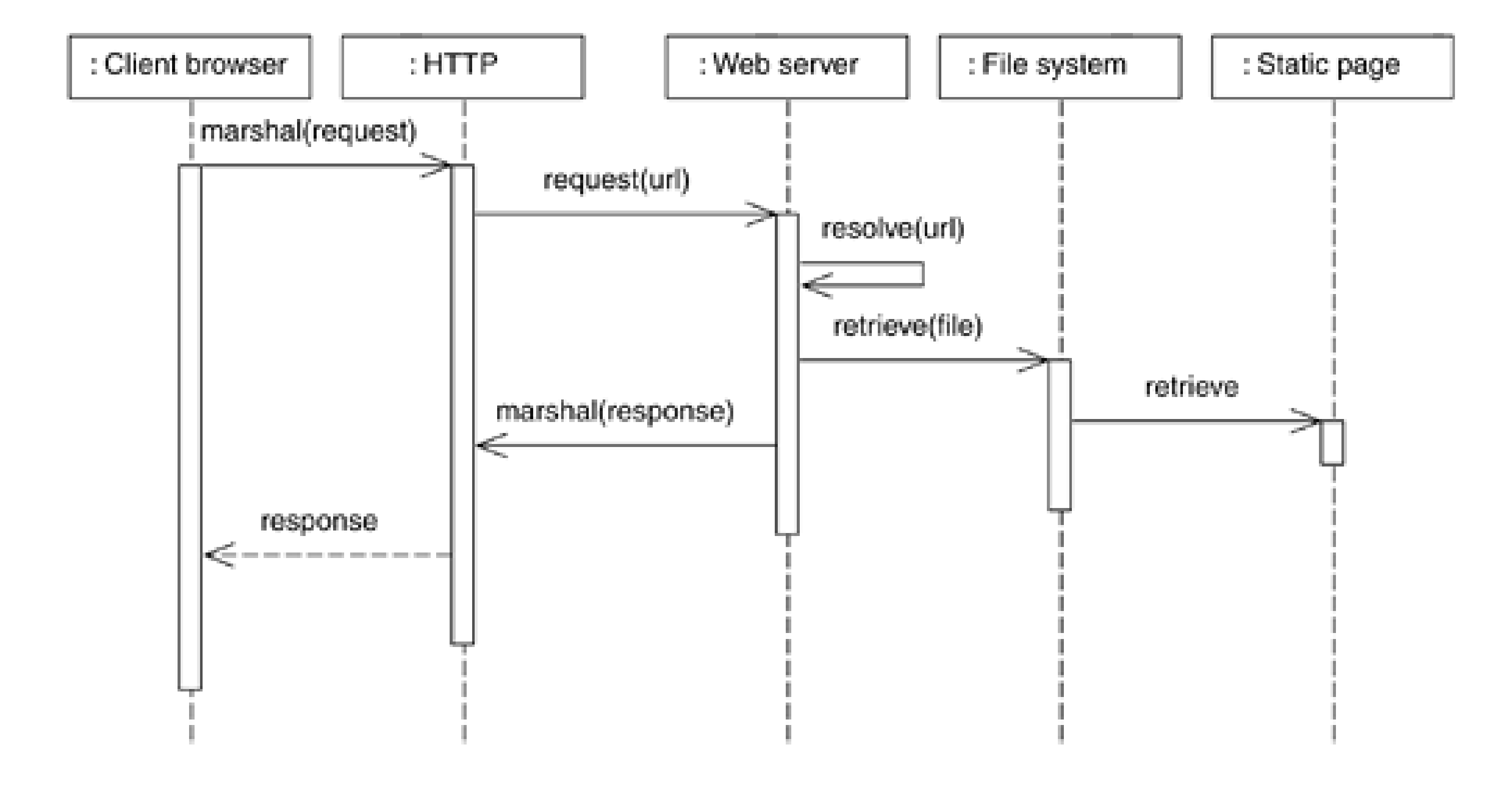
Design Viewpoint

- The major components exist on the server
- The major components are as follows:
  - Client browser: Any standard forms-capable HTML browser.
  - Web server: The principal access point for all client browsers.
  - HTTP connection: The most common protocol in use between client browsers and Web servers.
  - Static page: A Web page with UI and content information that does not go through any server-side processing.
  - Dynamic page: Web pages that go through some form of server-side processing.
  - Application server: The primary engine for executing server-side business logic.
  - Database server: The part of the system that maintains persistent state.
  - File system: For many Web applications, the file system is a first-class citizen of the architecture, because its directory structure often mirrors the logical URL structure.

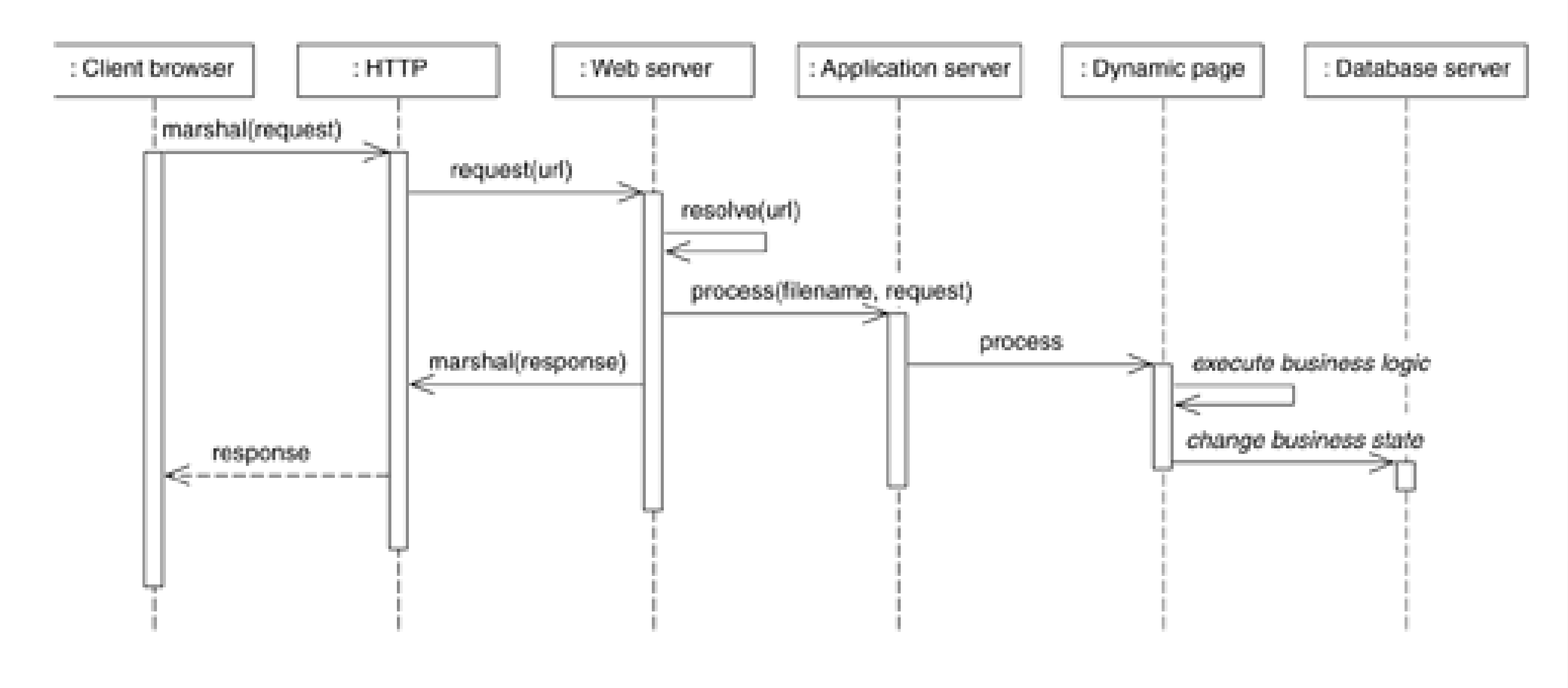
Principal participants in thin Web client collaboration



Static resource (page) request scenario



Dynamic resource (page) request scenario

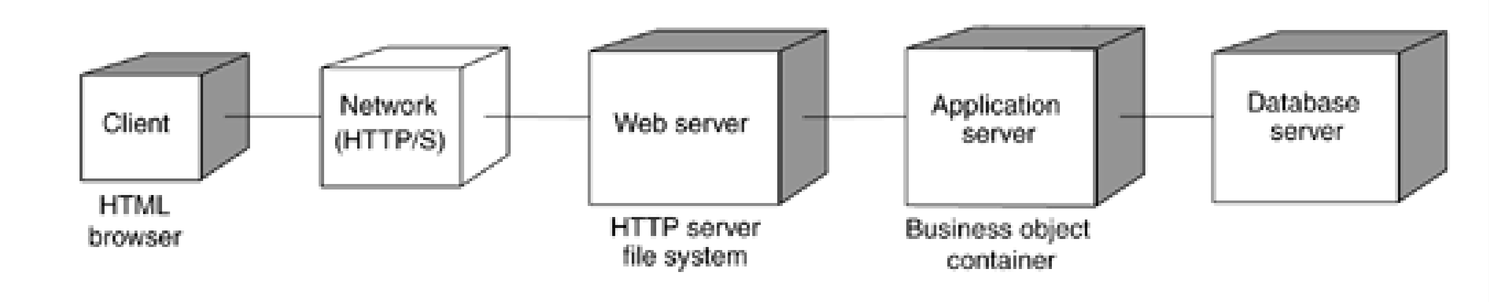


Thin Web Client

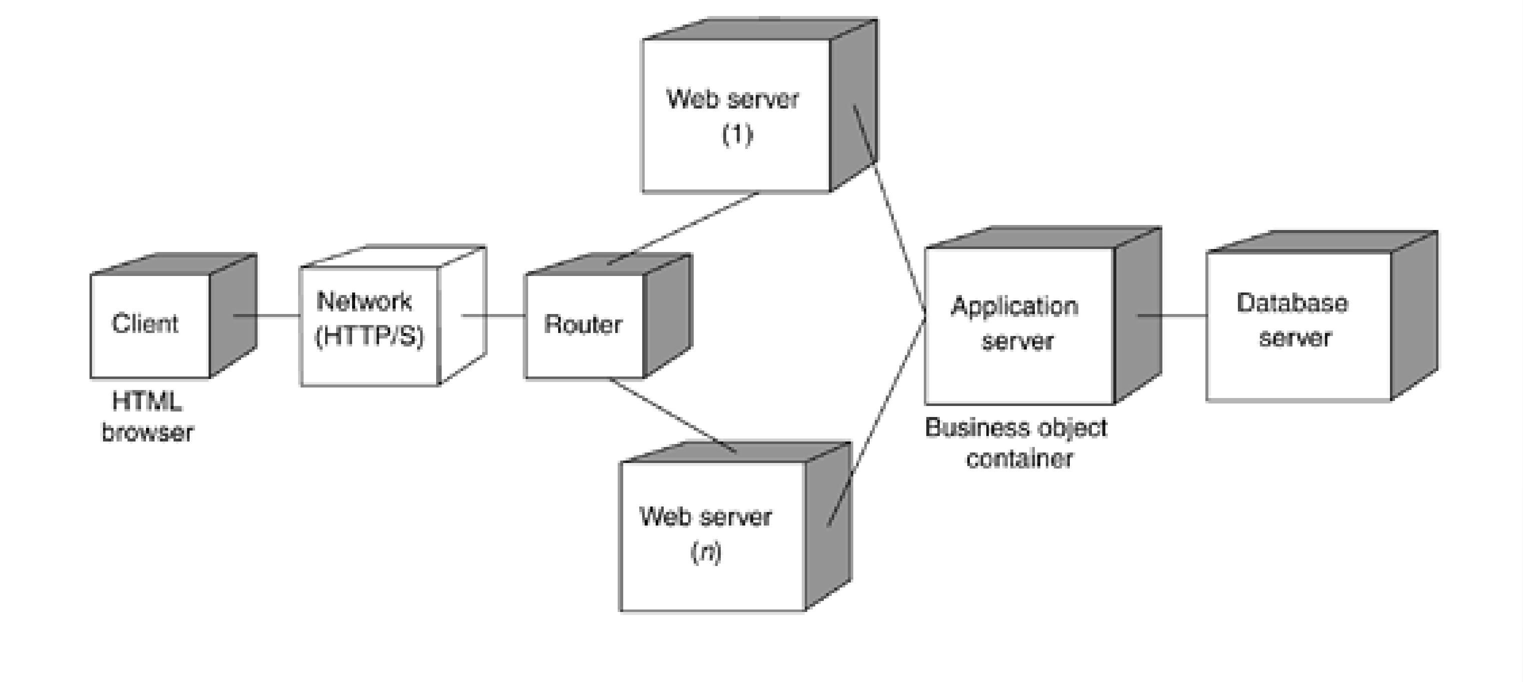
Realization Viewpoint

- Can vary greatly, depending on the scalability strategy.

Deployment of simple thin Web client application



Deployment of scalable thin Web client application



Thin Web Client

Test Viewpoint

- The major focus: to verify that the system meets the needs of the stakeholders, as specified by requirements.
- For Web applications, many areas that need testing may not be mentioned in the requirements.
  - Typically, these areas are a result of the system context, such as use on the Internet with unknown browser versions.
- For thin Web client architectures, key areas that need to be addressed during testing are
  - All supported browser versions
  - Firewall settings: use of only standard ports and protocols
  - Network latency
  - ISP and browser page caching
  - Cookie nature and use

Thick Web Client

- Extends the thin Web client pattern with client-side scripting and custom objects, such as ActiveX controls and Java applets
- The client can execute some of the business logic of the system and thus becomes more than a generalized user interface container
- Most appropriate for Web applications in which:
  - a certain client configuration and browser version can be assumed,
  - a sophisticated user interface is desired, and/or
  - a certain amount of the business logic can be executed on the client.

Thick Web Client (2)

Requirements Viewpoint

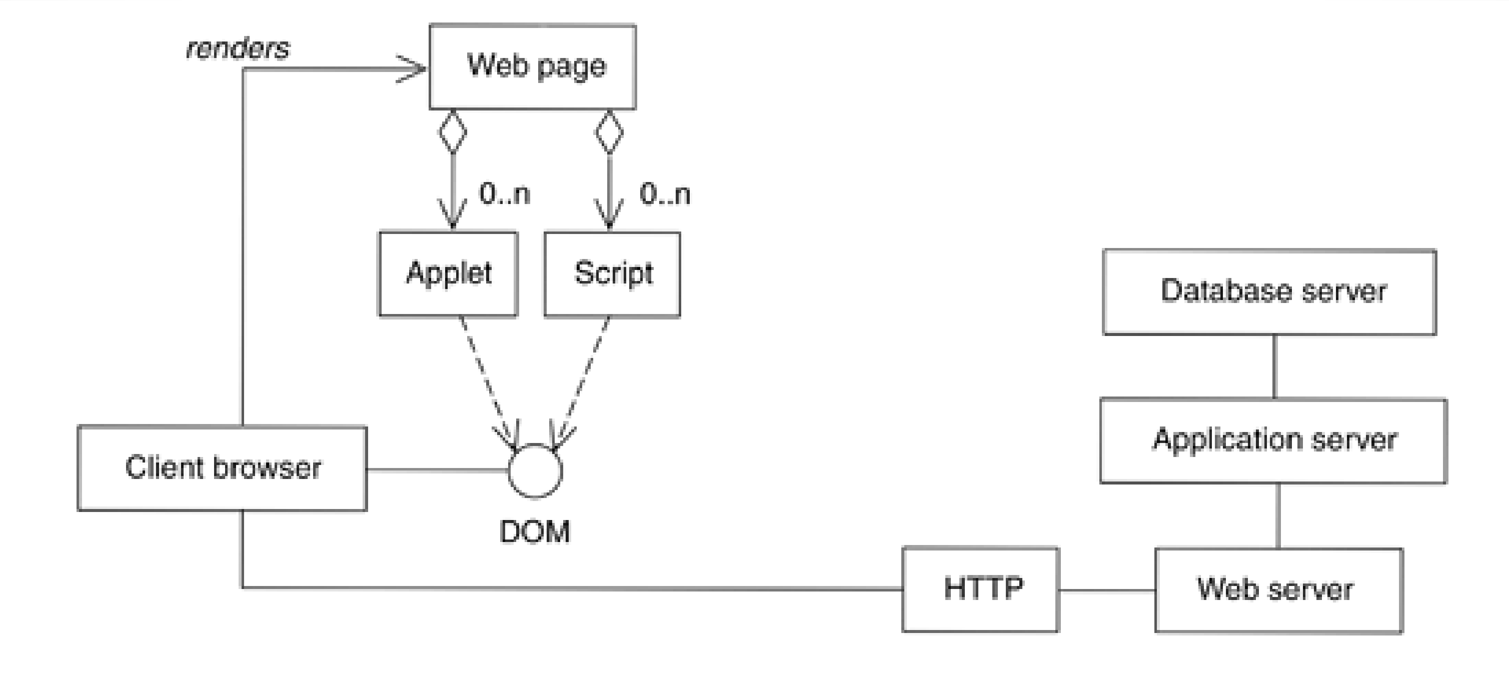
- The one additional new requirement: the ability to execute some of the system's logic on the client
- The most basic example: a script or an applet that validates a form's fields prior to submission to the server.
  - When possible, stopping invalid forms from being submitted to the server can improve the system's performance.
  - Additional scripts and controls on the client can also provide intelligent assistance in filling out complicated forms or in navigating the system's pages.
- The client's user interface experience can be enhanced with client-side activity
  - Sophisticated user interface controls, or scripting, can make the user interface easier to use or more attractive
  - Attractive interfaces are an important requirement in the tight e-commerce space, where the competition is only one click away

Thick Web Client (3)

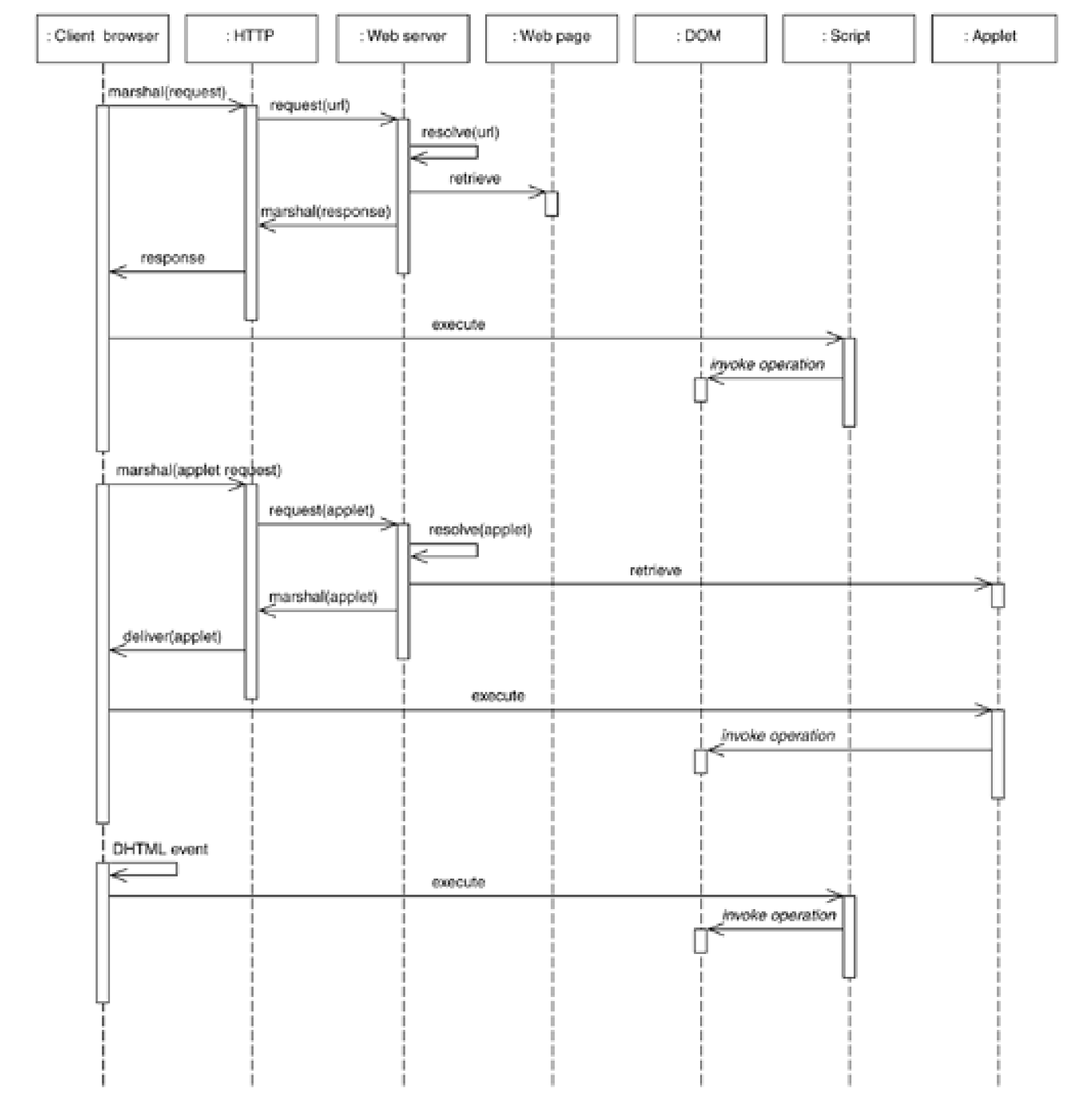
Design Viewpoint

- All communication between client and server, as in the thin Web client pattern, is done via HTTP or HTTPS. Because these protocols are "connectionless," most of the time, there is no open connection between client and server. Only during page requests does the client send information and requests to the server.
- This means that client-side scripting, ActiveX controls, and Java applets are limited to interacting with objects only on the client.
- Uses certain browser capabilities, such as Java applets or ActiveX controls, to execute business logic on the client.
- The principal components in the thick Web client architecture are the same as those in the thin Web client, with the addition of scripts, applets, and Document Object Model (DOM) interface supported by the client browser.

Principal participants in thick Web client architecture



Basic applet and script execution scenario



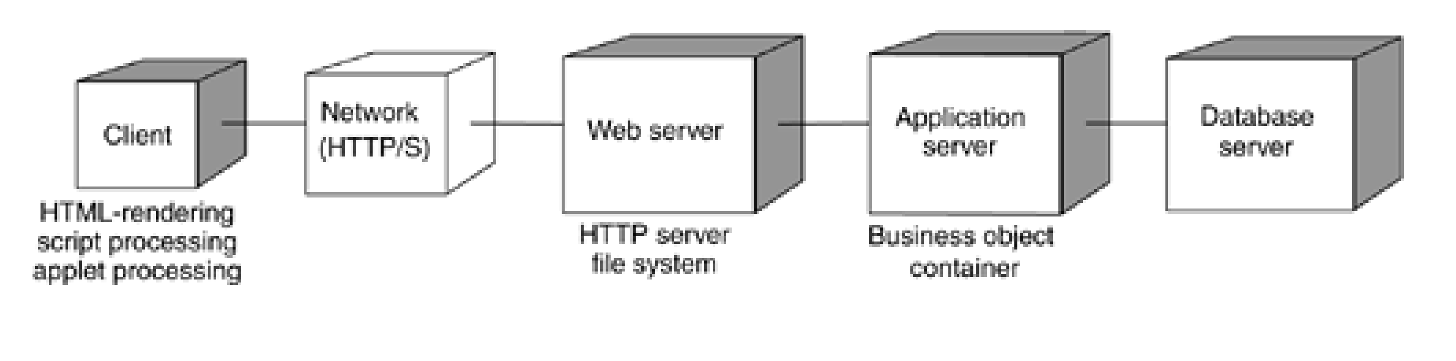
Thick Web Client (4)

Realization Viewpoint

- Nearly identical to that of the thin Web client
- The architectural's node:
  - client,
  - network,
  - Web server,
  - application server, and
  - database server.
- The only difference:

  the client node is responsible for processing scripts and executing applets

Deployment of simple thick Web client application



Thick Web Client (5)

Test Viewpoint

- Testing and quality assurance need to address a few additional areas:
  - Variations in vendors' scripting engines
  - Access to support Java classes or libraries
  - Network latency and its effects on multithreaded script processing
  - Careful examination of fault situations, or missing applet components
- One situation that developers often overlook is the system's behavior when certain resources, such as separate frames or applets, don't get loaded properly
  - This happens quite often in applications that operate over the Internet
  - The testing team needs to ensure that the system behaves gracefully when applets don't load in time

Web Delivery

- The Web is used primarily as a delivery mechanism for an otherwise traditional distributed object client/server system
- From one viewpoint, this type of application is a distributed object client/server application that happens to include a Web server and client browser as significant architectural elements

Web Delivery (2)

Requirements Viewpoint

- HTTP can be a very limiting protocol for the most sophisticated applications. The overhead required by standard HTTP traffic would make it impossible for many real-time equipment-monitoring applications to function on the Internet.
- Sometimes, good old-fashioned remote procedure calls or remote operation calls are the best solution.
- Most appropriate when there is significant control of client and network configurations.
- The greatest strength: its ability to leverage existing business objects in the context of a Web application.

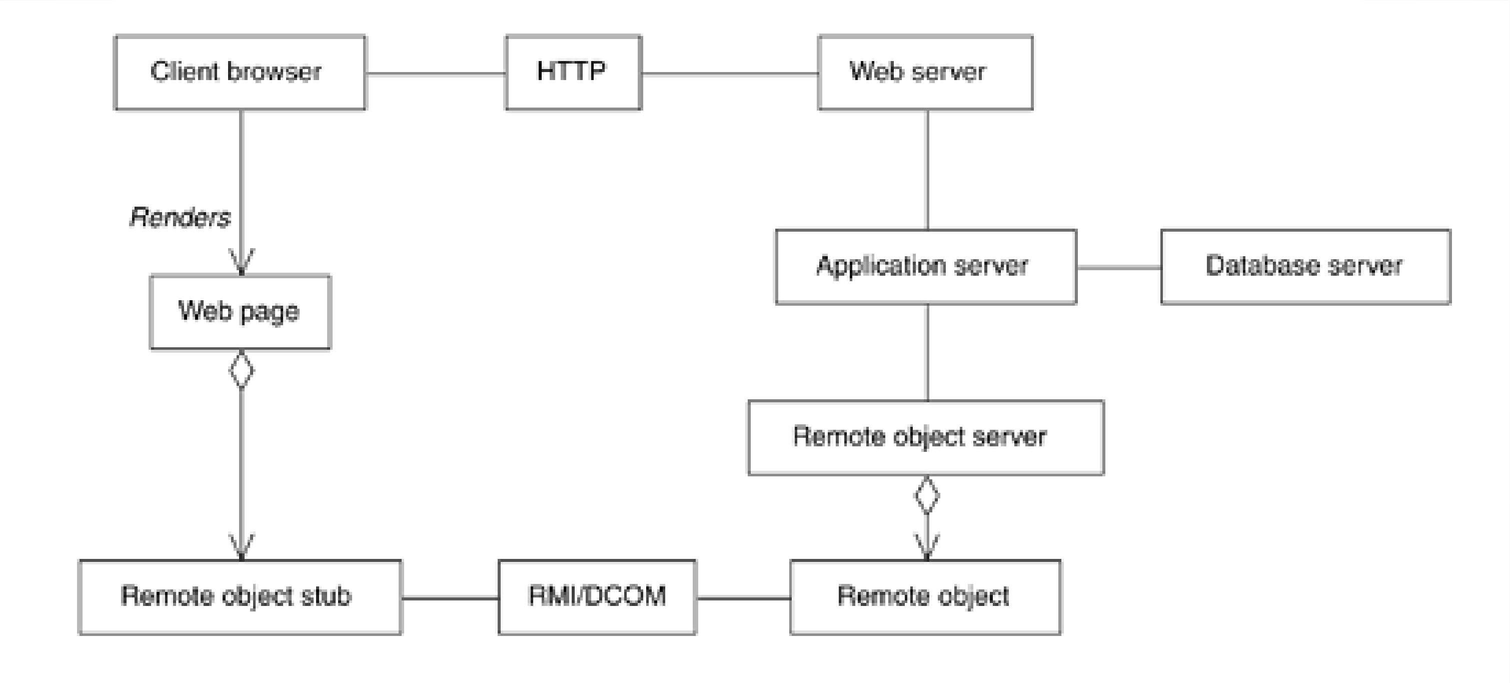
Web Delivery (3)

Design Viewpoint

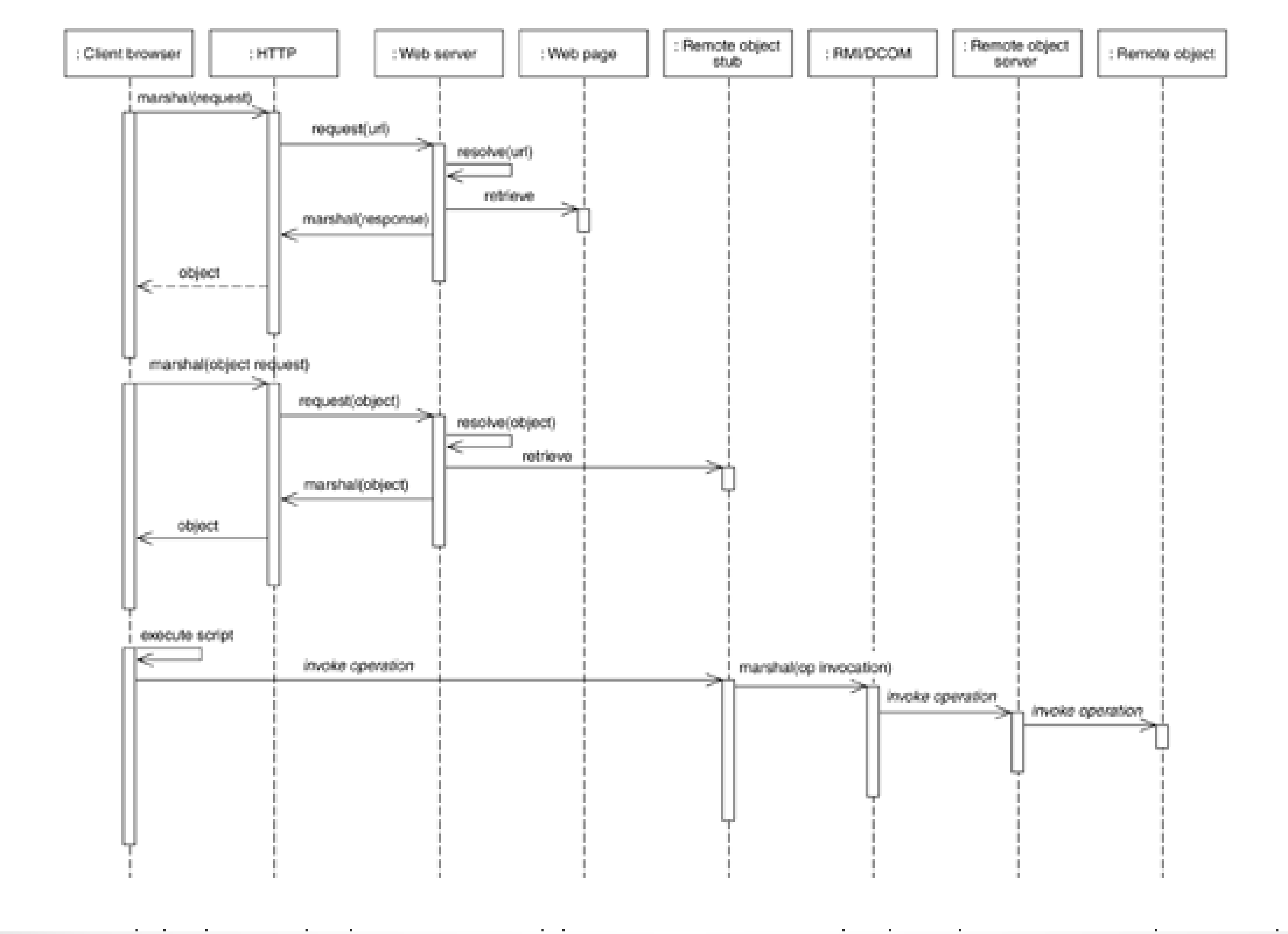
- The most significant difference between the Web delivery and the other Web application architecture patterns is the method of communication between the client and the server
  - In the other patterns, the primary mechanism is HTTP, a connectionless protocol that severely limits the designer when it comes to interactive activity between the user and the server
- The architecturally significant elements in the Web delivery pattern include all those specified in the two previous patterns, as well as the following:
  - Remote object/remote object stub: A conceptual object that can be interacted with, whose execution takes place on a remote machine.
  - Remote object server: A special server that hosts remote objects.
  - Remote object transfer protocol: The protocol that allows efficient access to remote objects over standard networks
    - Java RMI
    - DCOM

Principal participants

in the Web delivery architecture



Basic Web delivery remote object execution scenario

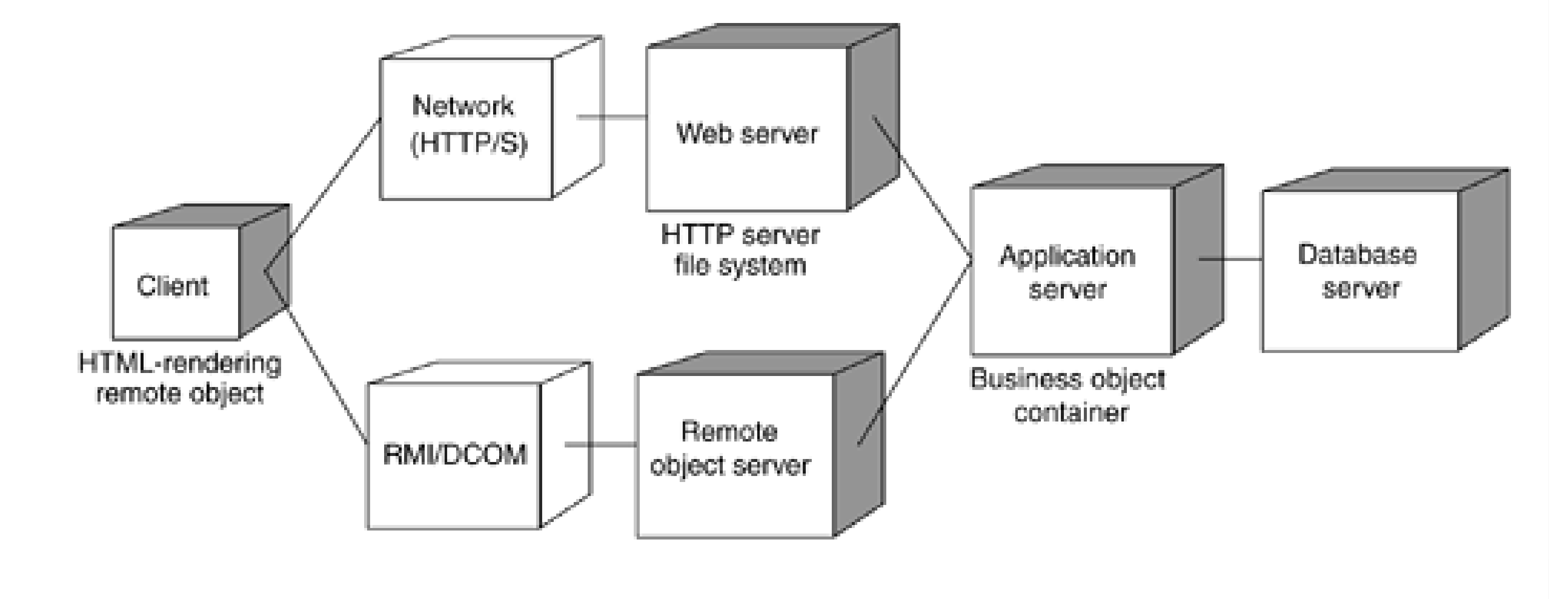


Web Delivery (4)

Realization Viewpoint

- Introduces a new node and device
  - The remote object server is responsible for making remote objects accessible to clients over the network
  - The protocol or method-invocation marshaling mechanism is the RMI or DCOM protocol
- It should be noted that in some situations, the application server and the remote object server might be one and the same in the realization viewpoint

Deployment of simple Web delivery application



Web Delivery (5)

Test Viewpoint

- Testing issues relevant to Web delivery architectures include careful monitoring of the infrastructure necessary to carry out the remote calls
- Because the need is often a result of the need for quick, long-lasting, and reliable communications directly with the server, careful examination of network latency and fault tolerance are important when testing these applications
- Testing and quality assurance must address a few additional areas:
    - The use of additional ports for remote object communication through firewalls
    - Performance over slow network connections
    - Effect on underpowered client machines

THANK YOU

Next Week

Please continue your final project report:

1. Use case Diagram
    - ➢ Determine use case diagram table for every use case (follow the format)

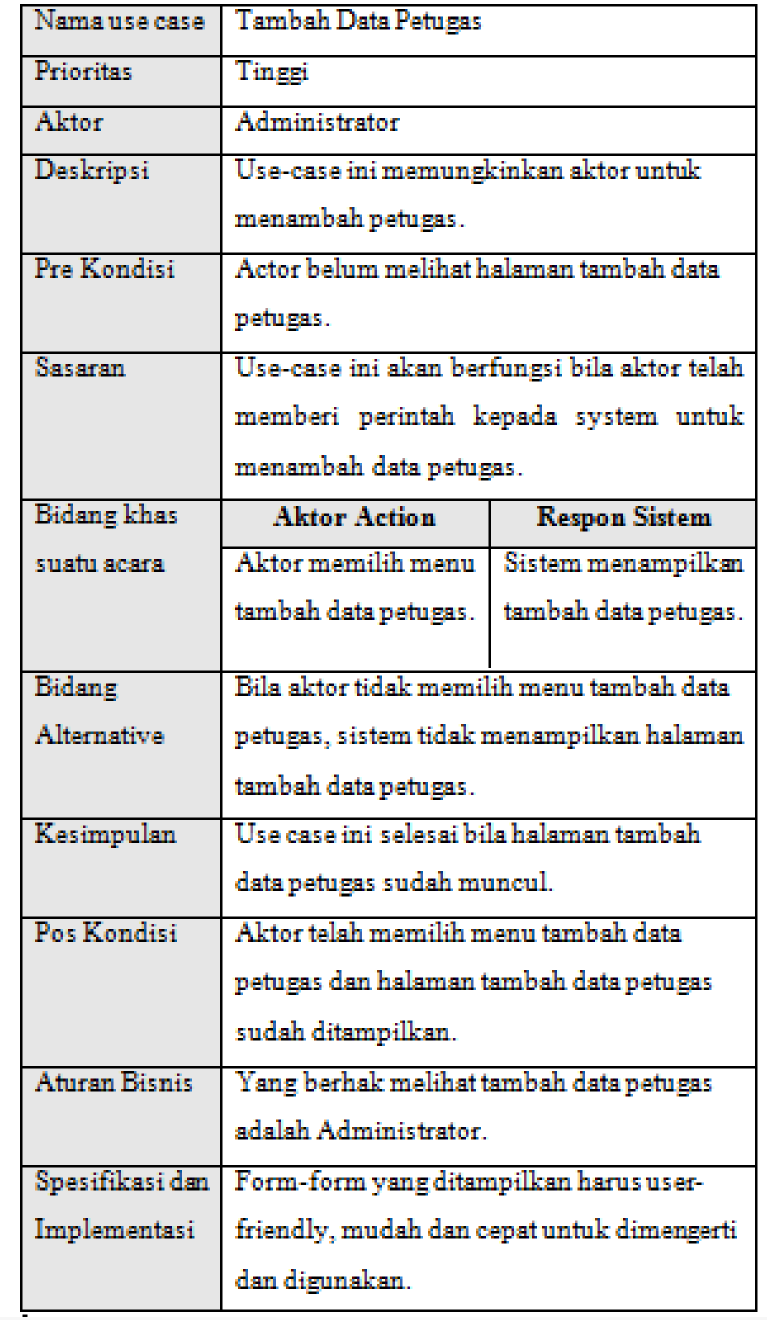
2. Class Diagram

3. Activity Diagram
    - ➢ Draw Activity diagram for every use case (10 use case = 10 activity diagram)

4. Sequence Diagram
    - ➢ Draw Sequence diagram for every use case (10 use case = 10 sequence diagram)

Draw with **Enterprise Architect**

Use Case Table Format



Nama use case	Tambah Data Petugas	
Prioritas	Tinggi	
Aktor	Administrator	
Deskripsi	Use-case ini memungkinkan aktor untuk menambah petugas.	
Pre Kondisi	Actor belum melihat halaman tambah data petugas.	
Sasaran	Use-case ini akan berfungsi bila aktor telah memberi perintah kepada system untuk menambah data petugas.	
Bidang khas suatu acara	**Aktor Action**	**Respon Sistem**
	Aktor memilih menu tambah data petugas.	Sistem menampilkan tambah data petugas.
Bidang Alternative	Bila aktor tidak memilih menu tambah data petugas, sistem tidak menampilkan halaman tambah data petugas.	
Kesimpulan	Use case ini selesai bila halaman tambah data petugas sudah muncul.	
Pos Kondisi	Aktor telah memilih menu tambah data petugas dan halaman tambah data petugas sudah ditampilkan.	
Aturan Bisnis	Yang berhak melihat tambah data petugas adalah Administrator.	
Spesifikasi dan Implementasi	Form-form yang ditampilkan harus user-friendly, mudah dan cepat untuk dimengerti dan digunakan.	