Course: Software Technologies and Enterprise Architecture

WEEK 8: Midterm Exam (Review Week 1- 7 materials)

Lemlem Kassa(Dr.)
Addis Ababa Science and Technology University (AASTU)

May, 2026

Week 8- Review Lectures form Week 1 to 7

Content

- Review Lectures (Week 1 - Week 7)
- Midterm Exam

Short Revision Week- 1 (Introduction to Enterprise Architecture)

1. System Architecture

- A **system** is a set of interconnected machines, applications, and network resources.
- **System architecture** visually represents how system components interact.

- It shows:
  - Main system functions
  - Connections between components
  - Information flow and responsibilities

Cont. … Short Revision Week- 1 (Introduction to Enterprise Architecture)

2. Enterprise Architecture (EA)

- EA is a **strategic practice** for aligning an organization's *strategy*, *processes*, and *technology*.
- Ensures the whole organization moves toward the same **vision and mission**.
- Helps design how a business should operate to achieve its goals.

Benefits of EA

- Aligns IT and business strategy
- Reduces risks
- Enhances efficiency and agility
- Supports digital transformation
- Helps optimize resources

Cont. ... Short Revision Week- 1 (Introduction to Enterprise Architecture)

EA Stakeholders

- Executives & business leaders
- IT teams & developers
- Project managers
- Security & compliance teams
- Operations and infrastructure teams

EA Domains

- Michael Platt's four perspectives:
  1. Business Architecture – business processes and standards
  2. Application Architecture – interactions between applications
  3. Information/Data Architecture – all organizational data
  4. Technology Architecture – hardware, OS, networks, tools

Cont. ... Short Revision Week- 1 (Introduction to Enterprise Architecture)

Challenges of EA

- Resistance to change
- Limited resources
- Complex IT environments
- Lack of standards
- Lack of clarity and governance
- Skills shortage

Cont. ... Short Revision Week- 1 (Introduction to Enterprise Architecture)

3. Solution Architecture

- More **tactical and project-focused** than EA.
- Solves specific business problems using available technologies.
- Defines **how a solution should be structured** within the enterprise.

Responsibilities of Solution Architects

- Identify/select technologies
- Create prototypes
- Understand business requirements
- Balance constraints (cost, time, quality)
- Collaborate with developers
- Create documentation
- Provide technical roadmaps

Cont. ... Short Revision Week- 1 (Introduction to Enterprise Architecture)

5. Business–IT Alignment

- Ensures IT strategy supports business strategy.
- IT is treated as a **strategic partner**, not just a support function.

6. Benefits

- Better decision making
- Efficient operations
- Cost savings
- Agility
- Improved customer experience

7. Key Steps to Align IT with Business

1. Understand business goals
2. Communicate clearly between IT and business
3. Prioritize IT projects that support strategy
4. Measure performance and outcomes
5. Adapt continuously

Short Revision (Week- 2– EA Frameworks)

1. What is an Enterprise Architecture Framework?

- A set of **guidelines, tools, and rules** used to design, plan, and manage an organization's IT systems so they align with business goals.

2. Types of EA Frameworks

1. **Comprehensive frameworks** – can be used in any organization (e.g., TOGAF).
2. **Industry frameworks** – designed for specific industries (e.g., FEAF for government).
3. **Domain frameworks** – focus on one area like data, business, or technology.

Cont. …Short Revision (Week 2 – EA Frameworks)

TOGAF (The Open Group Architecture Framework)

- A process-based framework that uses the Architecture Development Method (ADM) to build enterprise architecture.

Key Components

- **ADM (core)** – 10 phases guiding architecture development
- **Architecture Content Framework** – artifacts, diagrams, catalogs
- **Enterprise Continuum** – repository of architecture assets
- **Reference models** – TRM, III-RM
- **Capability Framework** – ensures EA can be practiced effectively

Cont. …Short Revision (Week 2 – EA Frameworks)

ADM Phases (very short)

- **Preliminary:** Prepare and set principles
- **Phase A:** Define architecture vision
- **Phase B:** Business architecture
- **Phase C:** Data + Application architecture
- **Phase D:** Technology architecture
- **Phase E:** Opportunities & solutions
- **Phase F:** Migration planning
- **Phase G:** Implementation governance
- **Phase H:** Change management
- **Requirements Management:** continuous throughout

Cont. …Short Revision (Week 2 – EA Frameworks)

Zachman Framework

- A **classification** framework (not a process).
  Organizes architecture using a **6×6 matrix**:

- **Rows = Perspectives**
  - Planner → Owner → Designer → Builder → Sub-contractor → User
- **Columns = What, How, Where, Who, When, Why**
  - (covering data, processes, locations, people, timing, goals)
- **Purpose:**
  - To provide a complete view of the enterprise from *every* angle.

Cont. …Short Revision (Week 2 – EA Frameworks)

FEAF (Federal Enterprise Architecture Framework)

- Used by the **US Federal Government**
- Combines ideas from **TOGAF + Zachman**
- Uses **5 reference models**: BRM, SRM, CRM, TRM, DRM
- Provides a standard way to design government IT systems

TOGAF vs Zachman

TOGAF	Zachman
A methodology (step-by-step process)	A classification schema (structure)
Tells you **how** to build EA	Shows **what** to document
ADM phases	6×6 matrix
More practical	More conceptual

Short Revision (Week 3 – SOA & Visualization)

1. What is SOA?

- Service-Oriented Architecture (SOA) is an architectural approach where software is built using **reusable, independent services** that communicate over a network. These services can be used across different applications in an enterprise.

2. Key Components of SOA

- **Service** – A reusable business function (e.g., payment service).
- **Service Provider** – Creates and hosts the service.
- **Service Consumer** – Uses the service.
- **Service Locator / Broker** – Helps consumers find and access services.
- **Enterprise Service Bus (ESB)** – Middleware that routes, transforms, and manages communication between services.

Cont. ...Short Revision (Week 3 – SOA & Visualization)

3. SOA Protocols

- **SOAP** – XML-based, strict structure, heavier.
- **REST** – Lightweight, flexible, supports multiple data formats.

4. Core Principles of SOA

- **Interoperability** – Works across platforms and languages.
- **Loose Coupling** – Services have minimal dependency on each other.
- **Abstraction** – Implementation details are hidden.
- **Granularity** – Services should represent a clear, focused function.

Cont. ...Short Revision (Week 3 – SOA & Visualization

5. Advantages of SOA

- Flexibility
- Scalability
- Cost-effectiveness (reuse of services)
- Better system integration and agility

6. Disadvantages of SOA

- Complexity
- Performance overhead
- Security risks
- Maintenance effort
- ESB can become a single point of failure

Cont. ...Short Revision (Week 3 – SOA & Visualization)

7. Real-World Applications

- Enterprise systems
- Cloud computing
- E-commerce
- Healthcare systems
- Defense systems

8. SOA vs Microservices

- **SOA**: Enterprise-level, uses ESB, services may be larger, shared data.
- **Microservices**: Application-level, no ESB, smaller services, independent databases, better for cloud-native apps.

Short Revision (Week 4- Software Product Lines (SPL)

1. Software Product

- A software product is software developed to solve a specific problem. It is created through the software development process and may include software only or both hardware and software.

Types of Software Products

a) Generic Products

- Ready-made software for general users
- Sold in the open market
- Example: Microsoft Office, Google Chrome

b) Customized Products

- Developed for a specific customer or organization
- Tailored to unique business needs
- Example: Custom payroll system, custom CRM system

Cont. …Short Revision (Week 4- Software Product Lines (SPL)

2. Characteristics of Good Software Products

- A good software product should have:
  - Efficiency
  - Maintainability
  - Dependability
  - Timely delivery
  - Budget control
  - Functionality
  - Adaptability

Cont. …Short Revision (Week 4- Software Product Lines (SPL)

3. Software Product Line (SPL)

- A Software Product Line is a collection of related software products that share common features and core assets.

Main Goal of SPL

- Reuse common components
- Reduce development time and cost
- Improve software quality
- Manage variations among products

Example

- Different versions of a mobile banking app sharing the same core system but having different features.

Cont. …Short Revision (Week 4- Software Product Lines (SPL)

4. Core Concepts of SPL

a) Software Assets

- Reusable resources such as:
- Source code
- Architecture
- Documentation
- Test cases
- Requirements

b) Decision Model

- Defines optional and variable features of products.

c) Variation Points

- Places where software behavior can change depending on product needs.

d) Product Outputs

- Final products created from shared assets

Cont. …Short Revision (Week 4- Software Product Lines (SPL)

5. Software Reuse

- Software reuse means developing software using existing components.

Reusable Components

- Source code
- Design
- Interfaces
- Documentation
- User manuals

Advantages

- Saves time
- Reduces cost
- Improves productivity
- Improves software quality
- Requires fewer resources

Cont. …Short Revision (Week 4- Software Product Lines (SPL)

6. Binding Time

- Binding time is the stage when decisions about variation points are finalized.

Types of Binding Time

- Development time
- Build time
- Install time
- Startup time
- Runtime

Example:

A user selecting language settings during installation.

Cont. Short Revision (Week 4- Software Product Lines (SPL)

7. Software Asset Management (SAM)

- SAM is the process of managing software assets, licenses, costs, and compliance throughout the software lifecycle.

Functions of SAM

- Inventory management
- License management
- Compliance checking
- Cost optimization
- Security improvement

Best Practices

- Conduct regular audits
- Use SAM tools
- Optimize licenses
- Create clear policies

Short Revision (Week 5- Software Development Methodologies

1. Software Development Life Cycle (SDLC)

- SDLC is a structured process used to:
- Plan
- Design
- Develop
- Test
- Deploy
- Maintain software

Purpose of SDLC

- Ensures systematic software development
- Reduces errors and risks
- Improves software quality
- Saves time and cost
- Aligns software with business goals

Cont. …Short Revision (Week 5- Software Development Methodologies

2. Software Development Methodology

- A software development methodology is a framework or approach used to manage and organize software development activities.
- It provides:
  - Development guidelines
  - Step-by-step procedures
  - Collaboration among stakeholders
  - Risk management
  - Quality assurance

Cont. …Short Revision (Week 5- Software Development Methodologies)

3. Importance of Software Development Methodologies

a) Structural Framework

- Provides organized development phases.

b) Risk Management

- Helps identify and manage project risks early.

c) Quality Assurance

- Ensures software quality through standards and testing.

d) Efficient Resource Utilization

- Optimizes time, budget, and workforce.

e) Communication and Collaboration

- Improves teamwork among developers, testers, and clients.

f) Adaptability

- Allows software to adapt to changing requirements.

Cont. …Short Revision (Week 5- Software Development Methodologies

4. Types of Software Development Models

1. Agile Methodology

- Iterative and flexible approach
- Continuous improvement and feedback

Advantages

- High-quality output
- Adaptable to changes
- Fast delivery

Disadvantages

- Lack of clear deadlines
- Less documentation

Cont. …Short Revision (Week 5- Software Development Methodologies

2. Waterfall Model

- Linear and sequential model
- Each phase completed before the next starts

Advantages

- Easy to manage
- Clear stages and deadlines

Disadvantages

- Difficult to change requirements later

Cont. …Short Revision (Week 5- Software Development Methodologies

3. Scrum Methodology

- Agile-based framework
- Works in short cycles called sprints

Scrum Process

1. Product backlog
2. Sprint planning
3. Sprint execution
4. Review
5. Retrospective

Advantages

- Fast feedback
- Easy problem detection

Disadvantages

- Not suitable for very large projects
- Difficult for junior teams

Cont. …Short Revision (Week 5- Software Development Methodologies

4. V-Model

- Verification and Validation model
- Testing is linked with every development stage

Advantages

- Easy to understand
- Strong testing process

Disadvantages

- Not suitable for complex projects with changing requirements

5. Spiral Model

- Combines iterative and waterfall approaches
- Focuses on risk analysis

Advantages

- Handles risks effectively
- Supports requirement changes

Disadvantages

- Expensive for small projects

Cont. …Short Revision (Week 5- Software Development Methodologies

6. RAD (Rapid Application Development)

- Focuses on quick development using prototypes

Advantages

- Fast development
- Frequent testing reduces errors

Disadvantages

- Not suitable for low-budget projects

7. Prototyping Model

- Creates an early version of the system for user feedback

Advantages

- Helps understand customer needs
- Easy visualization

Disadvantages

- Frequent changes may slow development

Cont. …Short Revision (Week 5- Software Development Methodologies

8. Lean Development

- Focuses on minimizing waste and reducing cost

Advantages

- Cost-effective
- Faster development

Disadvantages

- Less documentation

9. DevOps

- Integrates development and operations teams

Advantages

- Faster software delivery
- Continuous improvement

Disadvantages

- Compatibility issues in cloud environments

Short Revision (Week 6- Enterprise Unified Process

1. Unified Process (UP)

- Unified Process (UP) is an object-oriented software development methodology based on UML and use cases.

Main Features of UP

- Iterative and Incremental development
- Use-case driven
- Architecture-centric
- Risk management
- Continuous feedback and validation

Importance of UP

- Breaks large projects into smaller iterations
- Improves communication and collaboration
- Supports continuous improvement
- Reduces project risks

Cont. …Short Revision (Week 6- Enterprise Unified Process)

2. Key Principles of Unified Process

a) Iterative and Incremental Development

- Software is developed in small repeated cycles called iterations.

Benefits

- Early software delivery
- Better quality
- Easier risk management
- Flexibility for changes
- Continuous stakeholder feedback

b) Use-Case Driven

- Development is based on user requirements represented through use cases.

Cont. ..Short Revision (Week 6- Enterprise Unified Process)

c) Architecture-Centric

- Focuses on creating a strong software architecture throughout development.

d) Risk Management

- Risks are identified and controlled early in the project.

e) Continuous Validation

- Testing and feedback happen continuously to ensure quality.

Cont. ..Short Revision (Week 6- Enterprise Unified Process)

3. Workflows in Unified Process

1. Requirements Workflow
2. Analysis and Design Workflow
3. Implementation Workflow
4. Test Workflow
5. Deployment Workflow
6. Configuration and Change Management
7. Project Management Workflow
8. Environment Workflow

4. Artifacts in Unified Process

- Artifacts are documents and work products created during development.

Examples

- Vision Document ,Use Case Model Design Model ,Implementation Model ,Test Plan ,Deployment Plan User Manual .Project Schedule

Cont. ..Short Revision (Week 6- Enterprise Unified Process)

5. Roles in Unified Process

Role	Responsibility
Project Manager	Manages project activities
Architect	Designs system architecture
Analyst	Gathers requirements
Designer	Creates system design
Developer	Writes code
Tester	Tests the software

6. Rational Unified Process (RUP)

- RUP is a structured implementation of UP developed using UML.

Characteristics

- Iterative
- Incremental
- Use-case driven
- Risk-focused
- Component-based

Cont. ..Short Revision (Week 6- Enterprise Unified Process)

7. Phases of RUP

1. Inception

- Defines project scope
- Identifies requirements and risks

2. Elaboration

- Refines requirements
- Defines architecture
- Reduces risks

3. Construction

- Coding and testing take place
- System is developed

4. Transition

- Software is deployed to users
- Beta testing and feedback collection

5. Production

- Maintenance and updates of the software

Cont. ..Short Revision (Week 6- Enterprise Unified Process)

8. Advantages of RUP

- Strong documentation
- Good risk management
- Component reuse
- Better software quality

Disadvantages

- Complex process
- Requires experts
- Difficult integration
- Resource-intensive

9. Best Practices of RUP

- Develop incrementally
- Handle requirements using use cases
- Use modular architecture
- Use UML diagrams
- Ensure software quality
- Manage software changes

Cont. ..Short Revision (Week 6- Enterprise Unified Process)

10. Enterprise Unified Process (EUP)

- EUP is an extension of RUP designed for large organizations.

Main Difference

- EUP adds:
  1. Production phase
  2. Retirement phase

Goal

- Manage the entire software lifecycle including maintenance and retirement.

Cont. ..Short Revision (Week 6- Enterprise Unified Process)

11. Phases of EUP

1. Inception
2. Elaboration
3. Construction
4. Transition
5. Production
6. Retirement

12. Retirement Phase

The system is:

- Stopped from maintenance
- Archived
- Replaced or shut down
- Data migrated if needed

Short Revision (Week 7- Agile Architecture)

Overview of Agile Architecture

- **Iterative Approach:** Agile is a quick and adaptable development model that divides projects into small sections called "Time Boxes," typically lasting one to four weeks.
- **Concurrent Operations:** Unlike the sequential Waterfall model, development and testing occur simultaneously in agile.
- **Key Phases:** The lifecycle includes requirement gathering, design, iterative development, testing, deployment, and a final review/feedback stage.
- **Architectural Principles:** Emphasizes building scalable systems and maintaining high quality through practices like Extreme Programming (XP) and continuous refactoring.
- **Strategic Balance:** Architecture must be robust enough to be functional but flexible enough to evolve as requirements change.

Cont. ..Short Revision (Week 7- Agile Architecture)

Agile Project Planning

- **Planning Steps:** The process follows five key stages: Project Vision, Roadmap, Release Planning, Iteration Planning, and Daily Planning.
- **Vision & Roadmap:** Focuses on establishing long-term strategy aligned with organizational goals and breaking these down into high-level milestones.
- **Short-Term Execution:** Iteration planning zooms into 2–4 week sprints, while Daily Stand-Ups (Daily Scrums) synchronize activities for the next 24 hours

Cont. ..Short Revision (Week 7- Agile Architecture)

Agile and Enterprise Architecture (EA)

- **Modern Frameworks:** Organizations are increasingly adopting frameworks like the Scaled Agile Framework (SAFe) or the Agile EA Framework (AEAF) to meet modern business needs.
- **EA Pillars:** Focuses on Alignment (strategy to operations), Insight (organizational state), and Quality (solution improvement).
- **Collaborative Roles:** The model defines roles such as the Agile Lead Architect and Enterprise Architects who work closely with development teams.

Cont. ..Short Revision (Week 7- Agile Architecture)

Applications and Evaluation

- **Best Use Cases:** Agile is ideal for complex projects with unclear or shifting requirements and a strong need for customer involvement.
- **Advantages**: Offers rapid delivery, high customer satisfaction, and the ability to change requirements at any stage.
- **Disadvantages:** Can lead to higher costs, insufficient documentation, and may not be suitable for very small-scale projects.

Midterm Exam

Midterm Exam (Total Mark:15%)

Instruction :

This midterm exam contains 13 questions.

It contains three types of questions with a total point 15%

1. Multiple choice (5pts)
2. True or False (5pts)
3. Short Answer (5pts)

Students must to do all questions and put their answer properly on the space provided.

--Good Luck--

Mid-Exam (Total Mark:15%)

Part I: Multiple Choice Questions (1pt each)

_____1. What is the core objective of a Customer Relationship Management (CRM) system?

A. To manage internal employee payroll and benefits.

B. To manage company interactions with current and potential customers to improve profitability.

C. To automate the manufacturing production line.

D. To replace the company's primary website.

Mid-Exam (Total Mark:15%)

____2. Which of the following is identified as a "Back-Office" system?

A. CRM

B. Social Media Marketing

C. ERP (Enterprise Resource Planning)

D. Customer Support Portal

____3. Which of the following is considered a disadvantage of the Agile methodology?

A. Slow delivery of the final product

B. Difficulty in changing requirements late in the project

C. Potential for insufficient documentation

D. Low customer involvement

Mid-Exam (Total Mark:15%)

_____4. Which model is more suitable when requirements frequently change?

A. Agile
B. Spiral
C. Waterfall
D. V-Model

____5. Which technique identifies waste in business processes?

A. UML Modeling
B. Coding Review
C. Value Stream Mapping
D. Risk Analysis

Cont. .. Mid-Exam (Total Mark:15%)

Part II. True or False (1pt each)

1. Business Architecture is not reusable. _____

2. Under the Agile model, development and testing operations are performed concurrently._____

3. Agile architecture should be "heavy" and detailed at the start of the project to prevent future changes._____

4. The Daily Stand-Up (Daily Scrum) is typically a 15-minute meeting to synchronize team activities._____

5. EUP includes post-deployment support._____

Cont. …Mid-Exam (Total Mark:15%)

Part III. Short answer question (5pts)

Q1. Explain the phases of Enterprise Unified Process (EUP)?(3pts)._____

_____

Q2. Why is EUP considered more suitable for large organizations?(2pts)_____

_____


Thank you!