BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
Second Semester 2024-2025

Digital Learning Handout

Part A: Content Design

Course Title	Software Architectures
Course No(s)	SE* ZG651 / SS* ZG653
Credit Units	5
Credit Model	3-1-1
Course Author	Prof. Nayan Khare
Version No:	2.0
Date:	25/02/2025

Course Description:

Systems engineering and software architectures; Hatley-Pirbhai architectural template; architecture flow diagrams; requirements engineering and software architecture; architectural design processes; design post-processing; real-time architectures; architectural design patterns; software architecture and maintenance management; object oriented architectures; client-server architectures; forward engineering for object oriented and client-server architectures; emerging software architectures.

Course Objectives

No	Course Objective
CO1	To enable software engineers to architect software systems using industry best practices
CO2	To enable project managers to understand techniques of software architecture, and help them take appropriate decisions
CO3	To enable software professionals to take up research activities in the domain of software architecture

Text Book(s):

T1	Software Architecture in Practice, Third Edition, Len Bass, Paul Clements, Rick Kazman, Pearson 2013 ISBN:978-93-325-0230-7
T2	Essential Software Architecture, Second Edition, Ian Gorton, Springer 2011 ISBN:9783642191756

Reference Book(s) & other resources:

R1	Software Modelling and Design, Hassan Gomaa, Cambridge University Press 2011, ISBN:9780521764148
R2	Microsoft Application Architecture Guide, Second Edition, Microsoft 2009, ISBN: 9780735627109 [Availability: Online Free]
R3	Enterprise Architecture at Work: Modelling, Communication and Analysis, Third Edition, Marc Lankhorst et al., Springer 2013, ISBN:9783642296505
R4	Architecting for the cloud: Developing Multi-tenant Applications for the Cloud on Microsoft Windows Azure, Third Edition, Microsoft 2012, ISBN:978-1-62114-023-8 [Availability: Online Free]
R5	Architecting for the Cloud

	Amazon Web Services – Architecting for the Cloud: Best Practices, January 2011, Jinesh Varia [Availability: Online Free] https://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf DZone's Guide to Building and deploying applications on the cloud https://dzone.com/guides/building-and-deploying-applications-on-the-cloud
R6	Architecting for mobile ● https://magora-systems.com/mobile-app-development-architecture/ ● https://www.intellectsoft.net/blog/mobile-app-architecture/ ● https://www.uxpin.com/studio/blog/successful-mobile-applications-ui-design-patterns/ ● https://www.smashingmagazine.com/2018/02/comprehensive-guide-to-mobile-app-design/ Architecting Mobile Solutions for the Enterprise – Dino Esposito, 2012, Microsoft Press, ISBN: 978-0-7356-6303-2
R7	Identifying Architecturally Significant Functional Requirement Research paper by TCS – https://www.researchgate.net/publication/278242211_What_You_Ask_is_What_You_Get_Understanding_Architecturally_Significant_Functional_Requirements
R8	ATAM case study – Rockwell Collins – CAAS – Common Avionics Architecture System Video: https://youtu.be/da9MHLeTwvY Product description: https://www.rockwellcollins.com/Products_and_Services/Defense/Avionics/Integrated_Cockpit_Solutions/Common_Avionics_Architecture_System.aspx Rockwell Collins case study: https://resources.sei.cmu.edu/asset_files/TechnicalNote/2003_004_001_14150.pdf
R9	ATAM case study: Battlefield Control System: https://resources.sei.cm u.edu/asset_files/TechnicalReport/2000_005_001_13706.pdf
R10	Serverless architecture: ● https://docs.aws.amazon.com/lambda/latest/dg/welcome.html ● https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/serverless/web-app Container technology: https://www.cio.com/article/2924995/what-are-containers-and-why-do-you-need-them.html Caching: https://aws.amazon.com/caching/ , https://aws.amazon.com/caching/implementation-considerations/ Failure management in distributed systems: ● https://docs.microsoft.com/en-us/azure/architecture/guide/design-principles/self-healing ● https://dzone.com/articles/microservices-in-practice-1

R11	Technology topics Technologies: https://docs.microsoft.com/en-us/azure/architecture/ NoSQL databases https://www.dataversity.net/a-brief-history-of-non-relational-databases/# https://www.couchbase.com/resources/why-nosql https://www.thoughtworks.com/insights/blog/nosql-databases-overview Big data analytics Data mining & analytics: https://www.educba.com/data-mining-vs-data-analysis/ Technologies: https://www.edureka.co/blog/top-big-data-technologies/ Tools: https://www.guru99.com/big-data-analytics-tools.html Use cases: https://www.datamation.com/big-data/big-data-use-cases.html Case studies: https://data-flair.training/blogs/big-data-case-studies/ https://businessesgrow.com/2016/12/06/big-data-case-studies/ Hadoop https://www.mssqltips.com/sqlserverauthor/77/dattatrey-sindol/ https://en.wikipedia.org/wiki/Apache_Hadoop https://mapr.com/products/apache-hadoop/ https://www.sas.com/en_in/insights/big-data/hadoop.html Real time analytics https://www.sisense.com/glossary/real-time-analytics/ https://searchcustomerexperience.techtarget.com/definition/real-time-analytics https://www.scnsoft.com/blog/real-time-big-data-analytics-comprehensive-guide Spark https://spark.apache.org/streaming/ https://databricks.com/glossary/what-is-spark-streaming Use cases: https://www.qubole.com/blog/apache-spark-use-cases/ Machine learning https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/machine-learning-at-scale Primer: https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/machine-learning-primer-108796.pdf Steps: https://towardsdatascience.com/6-important-steps-to-build-a-machine-learning-system-d75e3b83686 Blockchain Introduction: https://www.pwc.co.uk/financial-services/fintech/assets/blockchain-an-intro.pdf Blockchain at Maersk: https://www.computerworld.com/article/3298522/ibm-maersk-launch-blockchain-based-shipping-platform-with-94-early-adopters.html

	Security OpenId: https://en.wikipedia.org/wiki/OpenID OAuth: https://tools.ietf.org/html/draft-ietf-oauth-use-cases-01#section-2.1 https://www.csoonline.com/article/3216404/what-is-oauth-how-the-open-authorization-framework-works.html De-militarized zone: https://searchsecurity.techtarget.com/definition/DMZ Firewall: https://www.cio.com.au/article/365101/top_seven_firewall_capabilities_effective_application_control/ https://www.fortinet.com/products/next-generation-firewall.html#services https://www.securedgenetworks.com/blog/11-Features-to-Look-for-in-Your-Next-Generation-Firewall LDAP: https://stackoverflow.com/questions/239385/what-is-ldap-used-for Integration strategies: Book 'Enterprise Integration Patterns' - Gregor Hohpe and Bobby Woolf IoT https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/iot/
R12	Technology trends: https://www.thoughtworks.com/radar https://www.infoq.com/ https://www.developertoarchitect.com/ Micro-frontends: https://martinfowler.com/articles/micro-frontends.html
R13	Transitioning from Developer to Architect: https://www.youtube.com/watch?v=JV8HNsFWHD4
R14	Case studies Architecture patterns – Case studies ● SoA at CIGNA ● SaleForce.com ● SoA at TripAdvisor ● Micro-Services at Danske Bank Case studies.zip Architecture evaluation and revision – Case study Scaling hospital call center Scaling, caching, reliability case study: Netflix

	http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html
R15	Microservices in practice: https://dzone.com/articles/microservices-in-practice-1
R16	Tactics to address different quality attributes: https://docs.microsoft.com/en-us/azure/architecture/patterns/category/availability

Learning Outcomes: Students will be able to

LO1	Ability to identify architecturally significant requirements and apply appropriate tactics to address them
LO2	Ability to determine appropriate architecture patterns for given requirements
LO3	Ability to document architecture that meets the needs of stakeholders
LO4	Ability to analyse architecture and determine its appropriateness given the requirement and determine risks
LO5	Awareness of best practices in design of cloud based applications, distributed applications and mobile applications
L06	Awareness of new technologies and their architecture and understanding of situations when to use these technologies
LO7	Ability evaluate the cost and benefit of different architecture options to aid in decision making

Modular Content Structure

1. Introduction to Software Architecture

- What is Software Architecture?
- Definitions of Software Architecture
- Architecture Structure and Patterns
- Good architecture
- Importance of Software architecture
- Contexts of Software architecture
- Architecture competence

2. Software Quality Attributes

- Understanding Quality Attributes
- Interoperability
- Testability
- Usability
- Performance
- Scalability
- Modifiability
- Security
- Availability
- Integration
- Other Quality Attributes

- Design Trade-Offs

3. Capturing Architecturally Significant Requirements

- Challenges in identifying ASRs
- Quality attribute Workshop
  - Understanding business goals from Sponsors
  - Identifying architectural drivers
  - Understanding Scenarios for each architectural driver via brainstorming with stakeholders
  - Prioritizing scenarios
  - Building a Utility tree
- Architecture design
  - Design strategy
  - Steps of Attribute-Driven design
  - Architecting in Agile projects

4. Documenting Software Architecture

- Importance of architecture documentation
- Architecture Views
- Quality attribute views – Security view, Communication view, Reliability view
- Combining Views
- Philippe Kruchten's 4+1 view
- Documentation Package
- Architectural templates - Hatley-Pirbhai

5. Layered architecture: Guidelines for different layers

- Presentation
- Business
- Data Layer
- Service
- Architecture evaluation (ATAM)
  - Factors for evaluation
  - Trade off analysis
  - Evaluation method
- Architecture Conformance techniques during implementation
- Architecture & Testing
- Architecture Reconstruction
  - Raw view extraction
  - View fusion
  - Finding violations
- Real-time architectures

6. Architectural patterns

- Layered
- MVC
- Publish-subscribe
- Pipe & Filter
- Service Oriented Architecture and Micro-services

7. Architectural patterns

- Broker
- Client server
- Peer-to-Peer
- Shared data
- Map-reduce
- Multi-tier
- Object-oriented architectures

8. Integration strategies and Architecting for Cloud

- File transfer, Messaging, RPC, WebSockets, API Gateways
- Architecting for Cloud
    - Benefits of Cloud based approach
    - Developing Multi-tenant Applications for the Cloud
    - Amazon Web Services tools
    - Trends in Cloud app development – languages, DB, Micro-services, CI / CD
- Technologies
    - Distributed Cache
    - Containers
    - Serverless architecture
- Failure management
    - CAP theorem
    - Failure management in distributed systems

9. Architecting for Mobile

- Types of mobile applications: native, cross platform, web app
- Design considerations
- Android Application components
- Patterns in Mobile Application
    - Store locally, sync later
    - Responsive design
    - UI design patterns

10. New technologies & their architecture

- Use cases and architecture of:
- Big data
    - NoSQL Databases
    - Hadoop
    - MapReduce
    - Real-time analytics
- Artificial intelligence & Machine Learning
- Block Chain
- IoT
- Security: AuthID, OAuth

11. Economic analysis of architectures

- Decision-making context
- Basis for economic analysis
- Cost Benefit Analysis Method

12. Recent developments and Emerging trends

- Web Assembly
- Service mesh
- Edge computing
- Forward Engineering for Object-Oriented and Client-Server Architectures

Part B: Learning Plan

Contact Session	List of Topic Title	Sub-Topics	Reference
1	**Introduction to Software Architecture**	● What is Software Architecture? ● Definitions of Software Architecture ● Architecture Structure and Patterns ● Good architecture ● Importance of Software architecture ● Contexts of Software architecture ● Architecture competence	T1 - 01, 02, 03, 24
2	**Software Quality Attributes**	● Understanding Quality Attributes ● Availability ● Performance ● Usability ● Security ● Modifiability	T1 - 04, 05, 06, 07, 08, 09, 10, 11, 12 R16
3	**Software Quality Attributes**	● Interoperability ● Testability ● Scalability ● Integration ● Other Quality Attributes ● Design Trade-Offs	R16
4	**Capturing Architecturally Significant Requirements**	● Challenges in identifying ASRs ● Quality attribute Workshop o Understanding business goals from Sponsors o Identifying architectural drivers o Understanding Scenarios for each architectural driver via brainstorming with stakeholders o Prioritizing scenarios o Building a Utility tree	R2, Chapter 4
5	**Architecture Design**	● Design strategy ● Steps of Attribute-Driven design ● Architecting in Agile projects	T1 - 15, 16, 17 R7
6	**Documenting Software Architecture**	● Importance of architecture documentation ● Architecture Views ● Quality attribute views – Security view,	T1 – 18 Lecture notes

		Communication view, Reliability view ● Combining Views ● Philippe Kruchten’s 4+1 view ● Documentation Package ● Architectural templates - Hatley-Pirbhai	
7	**Layered architecture: Guidelines for different layers**	● Presentation ● Business ● Data Layer ● Service ● Architecture evaluation (ATAM) o Factors for evaluation o Trade off analysis o Evaluation method	R2 T1 – 21 R8 R9
8	**Architecture: Conformance, Testing and Reconstruction**	● Architecture Conformance techniques during implementation ● Architecture & Testing ● Architecture Reconstruction o Raw view extraction o View fusion o Finding violations ● Real-time architectures	T1- 19 T1-20 Lecture notes
9	**Architectural patterns**	● Layered ● MVC ● Publish-subscribe ● Pipe & Filter ● Service Oriented Architecture and Micro-services	T1 R14
10	**Architectural patterns**	● Broker ● Client server ● Peer-to-Peer ● Shared data ● Map-reduce ● Multi-tier ● Object-oriented architectures	T1 R14 Lecture notes
11	**Integration Strategies and Architecting for Cloud**	● File transfer, Messaging, RPC, WebSockets, API Gateways ● Architecting for Cloud o Benefits of Cloud based approach o Developing Multi-tenant Applications for the Cloud o Amazon Web Services tools o Trends in Cloud app development – languages, DB, Micro-services, CI / CD	T1 R4 R5
12	**Technologies and Failure**	● Technologies o Distributed Cache	R10

	Management	o Containers o Server-less architecture ● Failure Management o CAP theorem o Failure management in distributed systems	
13	**Architecting for Mobile**	● Types of mobile applications: native, cross platform, web app ● Design considerations ● Android Application components ● Patterns in Mobile Application o Store locally, sync later o Responsive design o UI design patterns	R6
14	**New technologies & their architecture**	● Use cases and architecture of: ● Big data o NoSQL Databases o Hadoop o MapReduce o Real-time analytics ● Artificial intelligence & Machine Learning ● Block Chain ● IoT ● Security: AuthID, OAuth	R11, R12
15	**Economic analysis of architectures**	● Decision-making context ● Basis for economic analysis ● Cost Benefit Analysis Method	T1
16	**Recent developments and Emerging trends**	● WebAssembly ● Service mesh ● Edge computing ● Forward Engineering for Object-Oriented and Client-Server Architectures	R12 Lecture notes

Experiential Learning Components:

1. Lab work: 0
2. Project work: 0
3. Case Study: 5
4. Simulation: 0
5. Work Integrated Learning Assignment- 0
6. Design work/ Field work: 0

Objective of Experiential Learning Component:

Provide relevant case studies in Software Architecture.
Sample case studies include:

- SOA implementation at CIGNA
- Salesforce Architecture
- TripAdvisor Architecture
- Microservices implementation at Danske Bank
- Scaling, Caching, and Reliability case study at Netflix

Scope of Experiential Learning Component:
Scope is limited to demonstrating case studies related to software architecture.

Lab Infrastructure:
Not applicable

List of Experiments:
Not applicable

Evaluation Scheme:
Legend: EC = Evaluation Component; AN = After Noon Session; FN = Fore Noon Session

Evaluation Component	Name (Quiz, Lab, Project, Mid-term exam, End semester exam, etc.)	Type (Open book, Closed book, Online, etc.)	Weight	Duration	Day, Date, Session, Time
EC – 1*	Quiz (2 quizzes)	Online	10%	1 week	To be announced
	Assignment (2 assignments)	Online	20 %	20 days	To be announced
EC - 2	Mid-Semester Test	Closed Book	30%	2 hours	To be announced
EC - 3	Comprehensive Exam	Open Book	40%	2 ½ Hours	To be announced

EC1* (20% - 30%): Quiz (optional): 5-10 %, Lab Assignment/Assignment: 20% - 30%
Syllabus for Mid-Semester Test (Closed Book): Topics in Contact session: 1 to 8
Syllabus for Comprehensive Exam (Open Book): All topics

Important Links and Information:
eLearn Portal: https://elearn.bits-pilani.ac.in
Students must visit the eLearn portal regularly and stay updated with the latest announcements and deadlines.
Contact Sessions: Students should attend the online lectures as per the schedule provided on the eLearn portal.
Evaluation Guidelines:

1. EC-1 consists of either two Assignments or three Quizzes. Students will attempt them through the course pages on the eLearn portal. Announcements will be made on the portal in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted.

3. For Open Book exams: “open book” means text/ reference books (publisher copy only) and does not include any other learning material. No other learning material will be permitted during the open book examinations. For Detailed Guidelines refer to the attached document.
   EC3 Guidelines
4. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam, which will be made available on the eLearn portal. The Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the online lectures, and take all the prescribed evaluation components such as Assignments/Quizzes, Mid-Semester Tests and Comprehensive Exams according to the evaluation scheme provided in the handout.

****