BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES
COURSE HANDOUT

Part A: Content Design

Course Title	Software Architectures
Course No(s)	SE ZG651/SS ZG653
Credit Units	5
Course Author	Nayan Khare
Instructor –in-charge	HARVINDER SINGH JABBAL
Version No	1.4
Date	

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

Learning Outcomes:

No	Learning Outcome
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
LO6	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

Text Books:

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 Material:

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

Modular Content structure

Module No	List of Topic Title
M1	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
M2	Software Quality Attributes • Understanding Quality Attributes • Interoperability • Testability • Usability • Performance • Scalability • Modifiability • Security • Availability • Integration • Other Quality Attributes • Design Trade-Offs

M3	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
	
	Architecture design
	• Design strategy
	• Steps of Attribute-Driven design
	• Architecting in Agile projects

M4	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

M5	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

M6	Architectural patterns
	    • Layered
	    • MVC
	    • Publish-subscribe
	    • Pipe & Filter
	    • Service Oriented Architecture and Micro-services

M7	Architectural patterns
	    • Broker
	    • Client server
	    • Peer-to-Peer
	    • Shared data
	    • Map-reduce
	    • Multi-tier

M8	Integration strategies
	        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

M9.1	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

M9.2	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

M10.1	Economic analysis of architectures • Decision-making context • Basis for economic analysis • Cost Benefit Analysis Method
M10.2	Recent developments and Emerging trends • WebAssembly • Service mesh • Edge computing

Session plan

Session	List of Topic Title	Reference	Module
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	M1
2	Software Quality Attributes • Understanding Quality Attributes • Availability • Performance • Usability • Security • Modifiability	T1 - 04, 05, 06, 07, 08, 09, 10, 11, 12 R16	M2
3	Software Quality Attributes (cont.) • Interoperability • Testability • Scalability • Integration • Other Quality Attributes Design Trade-Offs	R16	M2

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
		T1 - 15, 16, 17
		 R7
			M3

5	Architecture design • Design strategy • Steps of Attribute-Driven design • Architecting in Agile projects	T1 - 15, 16, 17 R7	M3

6	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	T1 – 18	M4
7	Layered architecture: Guidelines for different layers • Presentation • Business • Data Layer • Service Architecture evaluation (ATAM) • Factors for evaluation • Trade off analysis • Evaluation method	R2 T1 – 21 R8 R9	M5
8	Architecture Conformance techniques during implementation Architecture & Testing Architecture Reconstruction • Raw view extraction • View fusion • Finding violations	T1 - 20 T1 - 19 T1 – 20	M5
9	Architectural patterns • Layered • MVC • Publish-subscribe • Pipe & Filter • Service Oriented Architecture and Micro-services	T1 R14	M6
10	Architectural patterns (cont.) • Broker • Client server • Peer-to-Peer • Shared data • Map-reduce • Multi-tier	T1 R14	M7

11	Integration strategies 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	T1 R4 R5	M8
12	Technologies • Distributed Cache • Containers • Server-less architecture Failure management • CAP theorem • Failure management in distributed systems	R10 R10	M8

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	M9.1

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
			M9.2

15	Economic analysis of architectures • Decision-making context • Basis for economic analysis • Cost Benefit Analysis Method	T1	M10

16	Recent developments and Emerging trends
	    • WebAssembly
	    • Service mesh
	    • Edge computing
		R12	M10

Mid sem exam syllabus: Modules 1 to 5
Compre exam syllabus: Modules 1 to 10

Contact sessions:

The contact session is expected to cover:

- Key concepts in the module
- Examples / case studies
- Experience sharing from participants
- Exercises

Students are expected to go through the reference material.

Students may be given home work at the end of each contact session.

Sample Assignments:

Assignment #1 (5% weight)

Objective: To get familiar with the software architecture basics.

Activity:

1. Choose an existing system from your workplace
2. Understand the purpose (goal) of the system & its key requirements
3. Study the architecture and understand the tactics used

Document your work in the following format in PPT:

1. Purpose of the system (Goal)
2. Key requirements of the system – functional & non-functional
3. Utility tree of Architecturally Significant Requirements (ASR)
4. Tactics used to achieve the top 5 ASRs
5. Software Architecture diagram – Context diagram, Module decomposition, Component & Connection diagram, Deployment diagram
6. Description of how the system works
7. Key learnings (one slide per participant)

Assignment #2 (10% weight)

Objective: To gain experience in architecting real life applications in domains such as Retail, Transportation, Healthcare, Hospitality, etc. Example systems: Swiggy, Uber, an IoT system to monitor health of industrial air conditioners.

Activity

1. Identify top 3 Architecturally Significant Requirements (ASRs) and write them in the form of a Utility tree. Why are these architecturally significant?
2. Describe in detail, the tactics you recommend for each ASR. For example, if caching is a tactic you recommend, please mention what you will cache, what tool you would use, how it will work, etc.

3. Draw 2 software architecture diagrams – component & connection view and deployment view – to understand how the system works.
4. Indicate important messages between components by labelling the connections in the C&C view. Also indicate the communication method used.
5. Draw sequence diagram for one major scenario (use case). Mention the scenario.
6. State the architecture patterns used. Explain, where in the architecture, these patterns have been used.
7. What did you learn by doing this assignment? Mention 3 key learnings. One slide per person.

Evaluation criteria:

a) Easy-to-understand diagrams
b) Clarity of description
c) Correctness of work products

Evaluation Components

No	Name	Type	Duration	Weight	Day, Date, Session, Time
EC-1	Quiz-I	Online		5%	September 1-10, 2024
	Quiz-II	Online		5%	October 10-20, 2024
	Quiz-III	Online		5%	November 1-10, 2024
	Assignment I	Online		5%	TBA
	Assignment-II	Online		10%	Individual Submission TBA Group Discussion TBA
EC-2	Mid-Semester Exam	Closed Book	2 Hours	30%	Saturday, 21/09/2024 (FN)
EC-3	Comprehensive Exam	Open Book	2 ½ Hours	40%	Saturday, 30/11/2024 (FN)

**Note** - Evaluation components can be tailored depending on the proposed model.

Syllabus for exams:

- Syllabus for Mid-Semester exam (Closed Book): Modules 1-5
- Syllabus for Comprehensive exam (Open Book): Modules 6-10

Evaluation Guidelines:

1. For Closed Book tests: No books or reference material of any kind will be permitted. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.

2. For Open Book exams: Use of prescribed and reference text books, in original (not photocopies) is permitted. Class notes/slides as reference material in filed or bound form is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.

3. 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. The genuineness of the reason for absence in the Regular Exam shall be assessed prior to giving permission to appear for the Make-up Exam. 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 lectures, and take all the prescribed evaluation components such as Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the handout.