

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	H S 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 <ul style="list-style-type: none"> • 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.cmu.edu/asset_files/TechnicalReport/2000_005_001_13706.pdf
R10	Serverless architecture: <ul style="list-style-type: none"> • 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

	<p>Caching: https://aws.amazon.com/caching/ , https://aws.amazon.com/caching/implementation-considerations/</p> <p>Failure management in distributed systems:</p> <ul style="list-style-type: none"> • https://docs.microsoft.com/en-us/azure/architecture/guide/design-principles/self-healing • https://dzone.com/articles/microservices-in-practice-1
R11	<p>Technology topics</p> <p>Technologies: https://docs.microsoft.com/en-us/azure/architecture/</p> <p>NoSQL databases</p> <p>https://www.dataversity.net/a-brief-history-of-non-relational-databases/#</p> <p>https://www.couchbase.com/resources/why-nosql</p> <p>https://www.thoughtworks.com/insights/blog/nosql-databases-overview</p> <p>Big data analytics</p> <p>Data mining & analytics: https://www.educba.com/data-mining-vs-data-analysis/</p> <p>Technologies: https://www.edureka.co/blog/top-big-data-technologies/</p> <p>Tools: https://www.guru99.com/big-data-analytics-tools.html</p> <p>Use cases: https://www.datamation.com/big-data/big-data-use-cases.html</p> <p>Case studies: https://data-flair.training/blogs/big-data-case-studies/</p> <p>https://businessgrow.com/2016/12/06/big-data-case-studies/</p> <p>Hadoop</p> <p>https://www.mssqltips.com/sqlserverauthor/77/dattatrey-sindol/</p> <p>https://en.wikipedia.org/wiki/Apache_Hadoop</p> <p>https://mapr.com/products/apache-hadoop/</p> <p>https://www.sas.com/en_in/insights/big-data/hadoop.html</p> <p>Real time analytics</p> <p>https://www.sisense.com/glossary/real-time-analytics/</p> <p>https://searchcustomerexperience.techtarget.com/definition/real-time-analytics</p> <p>https://www.scnsoft.com/blog/real-time-big-data-analytics-comprehensive-guide</p> <p>Spark</p> <p>https://spark.apache.org/streaming/</p> <p>https://databricks.com/glossary/what-is-spark-streaming</p> <p>Use cases: https://www.qubole.com/blog/apache-spark-use-cases/</p> <p>Machine learning</p> <p>https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/machine-learning-at-scale</p> <p>Primer: https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/machine-learning-primer-108796.pdf</p> <p>Steps: https://towardsdatascience.com/6-important-steps-to-build-a-machine-learning-system-d75e3b83686</p> <p>Blockchain</p> <p>Introduction: https://www.pwc.co.uk/financial-services/fintech/assets/blockchain-an-intro.pdf</p> <p>Blockchain at Maersk: https://www.computerworld.com/article/3298522/ibm-maersk-launch-blockchain-based-shipping-platform-with-94-early-adopters.html</p> <p>Security</p> <p>OpenId: https://en.wikipedia.org/wiki/OpenID</p> <p>OAuth: https://tools.ietf.org/html/draft-ietf-oauth-use-cases-01#section-2.1</p> <p>https://www.csoononline.com/article/3216404/what-is-oauth-how-the-open-authorization-framework-works.html</p> <p>De-militarized zone: https://searchsecurity.techtarget.com/definition/DMZ</p> <p>Firewall:</p> <p>https://www.cio.com.au/article/365101/top_seven_firewall_capabilities_effective_application_control/</p> <p>https://www.fortinet.com/products/next-generation-firewall.html#services</p> <p>https://www.securedgenetworks.com/blog/11-Features-to-Look-for-in-Your-Next-</p>

	<p>Generation-Firewall</p> <p>LDAP: https://stackoverflow.com/questions/239385/what-is-ldap-used-for</p> <p>Integration strategies:</p> <p>Book ‘Enterprise Integration Patterns’ - Gregor Hohpe and Bobby Woolf</p> <p>IoT</p> <p>https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/iot/</p>
R12	<p>Technology trends:</p> <p>https://www.thoughtworks.com/radar</p> <p>https://www.infoq.com/</p> <p>https://www.developertoarchitect.com/</p> <p>Micro-frontends: https://martinfowler.com/articles/micro-frontends.html</p>
R13	<p>Transitioning from Developer to Architect: https://www.youtube.com/watch?v=JV8HNsFWHD4</p>
R14	<p>Case studies</p> <p>Architecture patterns – Case studies</p> <ul style="list-style-type: none"> • SoA at CIGNA • SaleForce.com • SoA at TripAdvisor • Micro-Services at Danske Bank <p></p> <p>Case studies.zip</p> <p>Architecture evaluation and revision – Case study</p> <p></p> <p>Scaling hospital call center</p> <p>Scaling, caching, reliability case study: Netflix</p> <p>http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html</p>
R15	<p>Microservices in practice: https://dzone.com/articles/microservices-in-practice-1</p>
R16	<p>Tactics to address different quality attributes: https://docs.microsoft.com/en-us/azure/architecture/patterns/category/availability</p>

Content Structure

Module No	List of Topic Title	Reference	Recorded Lectures
M1	Introduction to Software Architecture <ul style="list-style-type: none"> • 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	RL 1.2 A Brief History of Software Architecture RL 1.3 Introduction to the Styles, Views and Three structures
M2	Software Quality Attributes <ul style="list-style-type: none"> • Understanding Quality Attributes • Interoperability • Testability • Usability • Performance • Scalability • Modifiability • Security • Availability • Integration • Other Quality Attributes • Design Trade-Offs 	T1 - 04, 05, 06, 07, 08, 09, 10, 11, 12 R16	RL 3.1 Quality classes, Quality attribute, quality attribute scenario and architectural tactics RL 4.1 Usability and its tactics RL 4.2 Availability RL 5.1 Modifiability RL 5.2 Performance RL 6.1 Security RL 6.2 Testability RL 6.3 Interoperability
M3	Capturing Architecturally Significant Requirements <ul style="list-style-type: none"> • Challenges in identifying ASRs • Quality attribute Workshop <ul style="list-style-type: none"> ○ 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 <ul style="list-style-type: none"> • Design strategy • Steps of Attribute-Driven design • Architecting in Agile projects 	T1 - 15, 16, 17 R7	RL 19.1 Architecture and Requirements RL 19.2 Designing the Architecture RL 8.2 Introducing Agile methodology
M4	Documenting Software Architecture <ul style="list-style-type: none"> • Importance of architecture documentation • Architecture Views • Quality attribute views – Security view, Communication view, Reliability view • Combining Views • <i>Philippe Kruchten's 4+1 view</i> • Documentation Package 	T1 – 18	RL 7.1 Introduction to OO Design RL 7.2 Introduction to UML RL 8.1 Documenting Architecture using UML RL 8.3 Rational Unified Process RL 20.1 Designing and Documenting the Architecture # 2

M5	Layered architecture: Guidelines for different layers <ul style="list-style-type: none"> • Presentation • Business • Data Layer • Service 	R2	Recording not available
	Architecture evaluation (ATAM) <ul style="list-style-type: none"> • Factors for evaluation • Trade off analysis • Evaluation method 	T1 – 21 R8 R9	
	Architecture Conformance techniques during implementation	T1 - 20	
	Architecture & Testing	T1 - 19	
	Architecture Reconstruction <ul style="list-style-type: none"> • Raw view extraction • View fusion • Finding violations 	T1 – 20	
M6	Architectural patterns <ul style="list-style-type: none"> • Layered • MVC • Publish-subscribe • Pipe & Filter • Service Oriented Architecture and Micro-services 	T1 R14	RL 9.1 Pattern Definition, Classification, Category and Intro to Layering RL 9.2 Layering Pattern RL 10.1 Pipe and Filter RL 10.2 Blackboard RL 11.1 Distributed System RL 12.1, 12.2 MVC Intro and detail RL 13.1, 13.2 Microkernel RL 13.3 Reflection
M7	Architectural patterns <ul style="list-style-type: none"> • Broker • Client server • Peer-to-Peer • Shared data • Map-reduce • Multi-tier 	T1 R14	

M8	<p>Integration strategies File transfer, Messaging, RPC, WebSockets, API Gateways</p> <p>Architecting for Cloud</p> <ul style="list-style-type: none"> • 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 <p>Technologies</p> <ul style="list-style-type: none"> • Distributed Cache • Containers • Serverless architecture <p>Failure management</p> <ul style="list-style-type: none"> • CAP theorem • Failure management in distributed systems 	<p>T1</p> <p>R4 R5</p> <p>R10</p> <p>R10</p>	<p>RL 17.1 Introduction and Virtualization basic</p> <p>RL 17.2 IAAS and Data storage</p> <p>RL 18.1 Quality attribute revisited</p> <p>RL 18.2 Multi-Tenant Architecture, Micro Services, CAP Theorem</p>
M9.1	<p>Architecting for Mobile</p> <ul style="list-style-type: none"> • Types of mobile applications: native, cross platform, web app • Design considerations • Android Application components • Patterns in Mobile Application <ul style="list-style-type: none"> ○ Store locally, sync later ○ Responsive design ○ UI design patterns 	R6	Recording not available
M9.2	<p>New technologies & their architecture</p> <p>Use cases and architecture of:</p> <ul style="list-style-type: none"> • Big data <ul style="list-style-type: none"> ○ NoSQL Databases ○ Hadoop ○ MapReduce ○ Real-time analytics • Artificial intelligence & Machine Learning • Block Chain • IoT • Security: AuthID, OAuth 	<p>R11 R12</p>	Recording not available
M10.1	<p>Economic analysis of architectures</p> <ul style="list-style-type: none"> • Decision-making context • Basis for economic analysis • Cost Benefit Analysis Method 	T1	
M10.2	<p>Recent developments and Emerging trends</p> <ul style="list-style-type: none"> • WebAssembly • Service mesh • Edge computing 	R12	

Contact sessions:

For each module there will be a contact session. 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 and / or recorded lectures, before coming to the class.

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%	February 13-23, 2023
	Quiz-II	Online		5%	March 20-30, 2023
	Assignment I	Online		5%	April 20-30, 2023
	Assignment-II	Online		10%	To be announced
EC-2	Mid-Semester Exam	Open Book	2 Hours	30%	Saturday, 11/03/2023 (FN)
EC-3	Comprehensive Exam	Open Book	2 ½ Hours	45%	Saturday, 20/05/2023 (FN)

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

Syllabus for exams:

- Syllabus for Mid-Semester exam (Open 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.