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

Part A: Content Design

Course Title	Software Quality Management
Course No(s)	MBA ZG661 / QM ZG662 / SS ZG661 / SE ZG661
Credit Units	4 [1 + 1 + 2 (Class Hours + Assignment + Student preparation)]
Course Author	K G Krishna
Version No	2.0
Date	

Course Objectives

No	Objective
CO1	Introduce Software Quality Management **as integral part of** Software Project Management employing Software Engineering Methodologies
CO2	Enhancing the **knowledge and skills** of engineers in adopting various Quality Assurance techniques in the development of large software systems or products.
CO3	Expose students to the **challenges of managing Quality** vs Cost/Schedule without compromising on customer satisfaction

Reference Books

R1	Software Quality Engineering – Jeff Tian, Wiley India, 2015
R2	Software Engineering: A Practitioner's Approach – Roger S. Pressman, 7th Edition, McGraw Hill, 2010
R3	Software Testing – A Craftsman's Approach – Paul C. Jorgensen, 4th Edition, CRC Press, 2014

Note:
There is no single recommended text book for this course. The suggested reference books are only indicative of the topics which get covered in them. The course is mostly delivered from a practitioner's perspective and hence no chapter-wise references could be provided. If students find the books hard to understand, they can substitute any good ref/text book/material they could find in their library or book-store or from web sources. Please note as this subject is (soft) and evolutionary, the terminology/standards adopted may vary from organization to organization. Students are strongly advised to refer to their organization's internal QMS documentation for comprehensive understanding of the subject
It's strongly recommended that students actively participate in online lectures and discussion-forums for demystifying the jargon to highlight the core concepts in Software Quality Management.

Module Content Structure

No	Title of the Module
M1: Introduction of Key Concepts of Quality, Quality Assurance and Quality Management	
	Introduction to Software Quality Assurance • Why is quality important? • Types of software & software projects • Key challenges in software development • Need for Quality culture in the organization • Impact of software development models on quality – Waterfall, Agile

	Software Quality Management – An Overview
	• Role of Quality management in Software engineering
	• Quality - a technical definition
	• Quality Concepts, Tools and Techniques
	• Software quality dimensions: Functional suitability, Reliability, Performance, Operability, Security, Maintainability, etc. (https://xbosoft.com/definition-software-quality/)
	• Quality Principles
	  o Quality is every one's responsibility
	  o Quality at every step
	  o Do it right first time
	  o What cannot be measured, cannot be improved
	  o Quality is a journey
	  o Doing the right thing and doing it right

	Demystifying Quality Concepts
	• Quality Assurance, Quality control, Quality Management, Quality Engineering
	• Role of process frameworks
	• Methodologies and tools adopted for Software Quality Engineering.
	  o Planning right project methodology, schedule, resources, risk mitigation
	  o Defining processes for development
	  o Training staff on engineering practices such as architecture, design, coding, testing, documentation
	  o Providing tools for different activities such as code analysis, testing, version management, release
	  o Doing review and testing
	  o Project monitoring & control
	  o Quality measurement and process improvement
	  o (https://www.javatpoint.com/quality-assurance)
	• Formal definitions of Quality by Standards groups of IEEE/ISO
	• Perspectives of Quality by Japanese and American Quality Gurus.

	Software Quality Processes • Concept of quality processes • QA activities • Testing, Reviews and Inspections • Continuous Improvement
M2: Software Quality Assurance: The Process & Activities	
	SQA Activities • Defect prevention and Defect detection • Quality Planning

	• Formal definition of ‘defect’, ‘error/bug’ • Role of defect measurement and analysis. • Techniques to prevent defects at different stages of software development • Configuration management concepts • Causal analysis and continuous improvement
	Software Testing • Testing strategies—white-box and black-box testing • Usage Based Testing and • Coverage Based Testing. • When to stop testing and start delivering • Partition Based Testing, • Testing based on Check-lists • Usage Based Statistical Testing • Coverage Based Testing
	Reviews & Inspections • Reviews • Inspections • Walkthroughs
	Fault-Tolerance • Fault-Tolerance techniques for critical systems
M3: Quality Planning, Metrics and QMS	
	Quality Planning in Projects • Typical format of Quality plan document • Quality goal • List of SQA activities to be planned • Project metrics to be tracked
	Quality Metrics and Baselining • Role of measurement and control in project management • SMART attributes for metrics • Concept of baselining • Leading & lagging indicators
	Product Metrics & Defect Propagation • Differences between Internal vs. External software attributes and their related metrics; • Problems of measurement; • Types of product metrics (Static vs. Dynamic) • Process of metrics measurement • Defect propagation • Cost of discovering defects late in the cycle
	Quality Management Systems • ISO 9001 model • CMMi model • Maturity levels of CMMi and its key process areas.
M4: QC Tools, Six-Sigma Methodology and Quality Culture	
	7 QC Tools, Six-Sigma Thinking • Concept of Quality control

	• 7 QC tools - Histograms, Pareto Charts, Cause and Effect Diagrams, Run Charts, Scatter Diagrams, Flow Charts, and Control Charts • Six sigma concept to reduce defects • DMAIC steps • Kaizen method for incremental improvement Quality Function Deployment (House of Quality)
	Cross-Cultural Issues in Quality Management • Importance of quality culture • Cultural differences in approach to quality - Indian, American and Japanese • Toyota management principles (https://flevy.com/blog/14-principles-of-lean-toyota-production-system-tps/)

Learning Outcomes:

No	Learning Outcomes
LO1	Students to be able to transition from their current role as coders/programmers to **developers of professional software systems** leveraging Organization's Quality Management Systems.
LO2	On-the-job demonstration of skills pertaining to **usage of techniques/tools** for planning and driving Quality Assurance in their projects
LO3	Ability to **plan and identify** right tools and activities within the constraints of the project

Part B: Contact Session Plan

Academic Term	Second Semester 2022-2023
Course Title	Software Quality Management
Course No	MBA ZG661 / QM ZG662 / SS ZG661 / SE ZG661
Lead Instructor	K.Venkatasubramanian
Instructor(s)	

Glossary of Terms

1. Contact Hour (CH) stands for a hour long live session with students conducted either in a physical classroom or enabled through technology. In this model of instruction, instructor led sessions will be for 22 CH.
    a. Pre CH = Self Learning done prior to a given contact hour
    b. During CH = Content to be discussed during the contact hour by the course instructor
    c. Post CH = Self Learning done post the contact hour
2. Contact Hour (CS) stands for a two-hour long live session with students conducted either in a physical classroom or enabled through technology. In this model of instruction, instructor led sessions will be for 11 CS.
    a. Pre CS = Self Learning done prior to a given contact session
    b. During CS = Content to be discussed during the contact session by the course instructor
    c. Post CS = Self Learning done post the contact session

3. RL stands for Recorded Lecture or Recorded Lesson. It is presented to the student through an online portal. A given RL unfolds as a sequences of video segments interleaved with exercises
4. SS stands for Self-Study to be done as a study of relevant sections from textbooks and reference books. It could also include study of external resources.
5. LE stands for Lab Exercises
6. HW stands for Home Work.
7. M stands for module. Module is a standalone quantum of designed content. A typical course is delivered using a string of modules. M2 means module 2.

Teaching Methodology (Flipped Learning Model)

The pedagogy for this course is centered around flipped learning model in which the traditional class-room instruction is replaced with recorded lectures to be watched at home as per the student's convenience and the erstwhile home-working or tutorials become the focus of classroom contact sessions. Students are expected to finish the home works on time.

Contact Session Plan

- Each Module (M#) covers an independent topic and module may encompass more than one Recorded Lecture (RL).
- Contact Sessions **(2hrs each week)** are scheduled alternate weeks after the student watches all Recorded Lectures (RLs) of the specified Modules (listed below) during the previous week
- In the flipped learning model, Contact Sessions are meant for in-classroom discussions on cases, tutorials/exercises or responding to student's questions/clarification--- may encompass more than one Module/RLs/CS topic.
- Contact Session topics listed in course structure (numbered CSx.y) may cover several RLs; and as per the pace of instructor/students' learning, the instructor may take up more than one CS topic during each of the below sessions.

Detailed Structure

Introductory Video/Document: *<< Introducing the faculty, overview of the course, structure and organization of topics, guidance for navigating the content, and expectations from students>>*

- Each of the sub-modules of **Recorded Lectures** (RLx.y) shall delivered via **30 – 60mins videos** followed by:
- **Contact session** (CSx.y) of 2Hr each for illustrating the concepts discussed in the videos with exercises, tutorials and discussion on case-problems (wherever appropriate); contact sessions (CS) may cover more than one recorded-lecture (RL) videos.

Course Contents

Contact Session 1

Time	Type	Description	Content

			Reference
Pre-CS			
During CS	CS1	Introduction to Software Quality Assurance • Why is quality important? • Types of software & software projects • Key challenges in software development • Need for Quality culture in the organization • Impact of software development models on quality – Waterfall, Agile	IPM
Post-CS			
Lab Reference			

Contact Session 2

Time	Type	Description	Content Reference
Pre-CS		RL1.1	

During CS	CS2	Software Quality Management – An Overview
		• Role of Quality management in Software engineering
		• Quality - a technical definition
		• Quality Concepts, Tools and Techniques
		• Software quality dimensions: Functional suitability, Reliability, Performance, Operability, Security, Maintainability, etc. (https://xbosoft.com/definition-software-quality/)
		• Quality Principles
		  ○ Quality is every one's responsibility
		  ○ Quality at every step
		  ○ Do it right first time
		  ○ What cannot be measured, cannot be improved
		  ○ Quality is a journey
		  ○ Doing the right thing and doing it right
			R1/C-1,2
			IPM

Post-CS			
Lab Reference			

Contact Session 3

Time	Type	Description	Content Reference
Pre-CS		RL1.2	
During CS	CS3	Demystifying Quality Concepts • Quality Assurance, Quality control, Quality Management, Quality Engineering • Role of process frameworks	R1/C-1,2 IPM

		• Methodologies and tools adopted for Software Quality Engineering.
		  ○ Planning right project methodology, schedule, resources, risk mitigation
		  ○ Defining processes for development
		  ○ Training staff on engineering practices such as architecture, design, coding, testing, documentation
		  ○ Providing tools for different activities such as code analysis, testing, version management, release
		  ○ Doing review and testing
		  ○ Project monitoring & control
		  ○ Quality measurement and process improvement
		  ○ (https://www.javatpoint.com/quality-assurance)
		• Formal definitions of Quality by Standards groups of IEEE/ISO
		• Perspectives of Quality by Japanese and American Quality Gurus.	

Post-CS			
Lab Reference			

Contact Session 4

Time	Type	Description	Content Reference
Pre-CS		RL1.3, RL2.1	
During CS	CS4	Software Quality Processes (Agile Methods) • Concept of quality processes • QA activities • Testing, Reviews and Inspections • Continuous Improvement SQA Activities • Defect prevention and Defect detection • Quality Planning • Formal definition of ‘defect’, ‘error/bug’ • Role of defect measurement and analysis. • Techniques to prevent defects at different stages of software development • Configuration management concepts • Causal analysis and continuous improvement	R2/C-2 R1/C-3,4 IPM
Post-CS			
Lab Reference			

Contact Session 5

Time	Type	Description	Content Reference
Pre-CS		RL2.2, RL2.3	
During CS	CS5	Software Testing • Testing strategies—white-box and black-box testing • Usage Based Testing and • Coverage Based Testing. • When to stop testing and start delivering • Partition Based Testing, • Testing based on Check-lists • Usage Based Statistical Testing • Coverage Based Testing	R1/C-6,7
Post-CS			
Lab Reference			

Contact Session 6

Time	Type	Description	Content Reference
Pre-CS		RL2.4, RL2.5	
During CS	CS6	Reviews & Inspections • Reviews • Inspections • Walkthroughs Fault-Tolerance • Fault-Tolerance techniques for critical systems	R1/C-14 R1/C-16
Post-CS			
Lab Reference			

Contact Session 7

Time	Type	Description	Content Reference
Pre-CS		RL3.1, RL3.2	
During CS	CS7	Quality Planning in Projects • Typical format of Quality plan document • Quality goal • List of SQA activities to be planned • Project metrics to be tracked Quality Metrics and Baselining	R2/C-24, 25 IPM

		• Role of measurement and control in project management • SMART attributes for metrics • Concept of baselining • Leading & lagging indicators	
Post-CS			
Lab Reference			

Contact Session 8

Time	Type	Description	Content Reference
Pre-CS		RL3.3, RL3.4	
During CS	CS8	Product Metrics & Defect Propagation • Differences between Internal vs. External software attributes and their related metrics; • Problems of measurement; • Types of product metrics (Static vs. Dynamic) • Process of metrics measurement • Defect propagation • Cost of discovering defects late in the cycle Quality Management Systems • ISO 9001 model • CMMi model • Maturity levels of CMMi and its key process areas.	R2/C-23 R2/C-30
Post-CS			
Lab Reference			

Contact Session 9

Time	Type	Description	Content Reference
Pre-CS		RL4.1	
During CS	CS9	7 QC Tools • Concept of Quality control • 7 QC tools - Histograms, Pareto Charts, Cause and Effect Diagrams, Run Charts, Scatter Diagrams, Flow Charts, and Control Charts	IPM
Post-CS			
Lab Reference			

Contact Session 10:

Time	Type	Description	Content Reference
Pre-CS		RL4.2	
During CS	CS10	Six-Sigma Thinking • Six sigma concept to reduce defects • DMAIC steps • Kaizen method for incremental improvement Quality Function Deployment (House of Quality)	IPM
Post-CS			
Lab Reference			

Contact Session 11:

Time	Type	Description	Content Reference
Pre-CS		RL4.3, RL4.4	
During CS	CS11	Cross-Cultural Issues in Quality Management • Importance of quality culture • Cultural differences in approach to quality - Indian, American and Japanese • Toyota management principles (https://flevy.com/blog/14-principles-of-lean-toyota-production-system-tps/)	IPM
Post-CS			
Lab Reference			

***Ref Book/Chap Ref**: At times the topics presented in the lecture-videos or PPTs may not have exact mapping to the cited references; student is expected to explore sources within the organization and on the web for up-to-date treatment of the topic

#R1, R2 (Reference Books); **IPM**: Instructor-provided Material;

Evaluation Scheme:

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

No	Name	Type	Duration	Weight	Day, Date, Session, Time
EC-1	Quiz-I	Online	-	10%	February 13-23, 2023
	Quiz-II	Online	-	10%	March 20-30, 2023
	Quiz-III	Online	-	10%	April 20-30, 2023
EC-2	Mid-Semester Test	Open Book	2 hours	30%	Saturday, 11/03/2023 (Evening)

EC-3	Comprehensive Exam	Open Book	2 ½ hours	40%	Saturday, 20/05/2023 (Evening)

Important Information:

Syllabus for Mid-Semester Test (Open Book): Topics in CS 1-5.
Syllabus for Comprehensive Exam (Open Book): All topics given in plan of study

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.

Appendix: A – Additional details about Recorded Lectures

Type	Title	Description
M1: Introduction of Key Concepts of Quality, Quality Assurance and Quality Management		
RL1.1	Software Quality Management – An Overview	**RL1.1 Software Quality Management – An Overview** This introductory module gives an overview of Software Quality by illustrating its role in Software Engineering. ▪ Quality - a technical definition ▪ Quality Concepts, Tools and Techniques
RL1.2	Demystifying Quality Concepts	**RL1.2 Demystifying Quality Concepts** This module demystifies various terms associated with quality—QA, QC, QM, Quality Engineering—as well as the role of process frameworks, methodologies and tools adopted for Software Quality Engineering. Also highlights formal definitions of Quality by Standards groups of IEEE/ISO along with various perspectives of Quality by Japanese and American Quality Gurus.
RL1.3	Software Quality Processes	**RL1.3 Software Quality Processes** Process is the bedrock of formal quality systems. This module introduces the concept of Quality Process and its associated Quality Assurance (QA) activities such as Testing, Reviews and Inspections with focus on "Continuous Improvement" which drives the overall maturity of Quality Management System in the organization.
M2: Software Quality Assurance: The Process & Activities		
RL2.1	SQA Activities	**RL2.1 SQA Activities** This module highlights the difference between defect prevention and defect detection activities. Presents the big picture of SQA encompassing Quality Planning and Continuous Improvement by giving formal definition of 'defect', 'error/bug' and the role of defect measurement and analysis.
RL2.2	Software Testing	**RL2.2 Software Testing** This module highlights various Testing strategies—white-box and black-box testing—introducing Usage Based Testing and Coverage Based Testing. Also discussed is the issue of 'when to stop testing and start delivering'.
RL2.3	Testing Techniques	**RL2.3 Testing Techniques** This modules continues previous discussion on Software Testing by describing Partition Based Testing, Testing based on Check-lists, Usage Based Statistical Testing and Coverage Based Testing, etc.
RL 2.4	Reviews & Inspections	**RL 2.4 Reviews & Inspections** "Prevention of defects is better than Testing for defects later" – is the spirit behind Reviews, Inspections and Walkthroughs. This modules highlights these important SQA activities—from formal reviews/inspections to little informal code-walkthroughs—adopted as part of most formal software development methodologies.
RL 2.5	Fault Tolerance	**RL 2.5 Fault Tolerance** In spite of rigorous SQA, certain undetected defects may percolate to the final system when delivered. When faults lurking in the built system cannot be tolerated at any cost for certain mission-critical applications/real-time systems, Fault-Tolerance techniques have to be employed to minimize or avoid catastrophic damage in the rare and unfortunate event of such a fault flaring up. This module highlights various Fault Tolerance (FT) techniques that can be

		designed for in the development of such critical systems.
M3: Quality Planning, Metrics and QMS		
RL3.1	Quality Planning in Projects	**RL3.1 Quality Planning in Projects** This module explains how Quality gets embedded into a Project by Quality Planning as part of Project Planning; Illustrates the typical format of Quality Plan document; List of various SQA activities to be planned along with Product and Project Metrics/Goals to be defined and tracked throughout Project Life-Cycle.
RL3.2	Quality Metrics and Baselining	**RL3.2 Quality Metrics and Baselining** "What we can't measure, we can't manage"—measurement is the key to management of quality. This module illustrates the role of measurement and control in project management; Defines various S.M.A.R.T attributes for a well-defined metric and the concept of metrics 'Baselining'.
RL 3.3	Software Product Metrics & Defect Propagation	**RL3.3 Product Metrics & Defect Propagation** This module illustrates differences between Internal vs. External software attributes and their related metrics; the problems of measurement; types of product metrics (Static vs. Dynamic) and the process of their measurement;
RL 3.4	Quality Management Systems	**RL 3.4 Quality Management Systems** QMS drives organization's quality culture and processes. This module describes two common models of Quality Systems adopted in a typical Indian IT services organization—ISO 9001 and CMMi; describes maturity levels of CMMi and its key process areas.
M4: QC Tools, Six-Sigma Methodology and Quality Culture		
RL4.1	Seven QC Tools	**RL4.1 7 QC Tools** Quality Control is the concept (precursor to Quality Assurance) that emerged during manufacturing era in the late 20th century. This module gives an overview of the popular 7 visual QC tools developed by Kaoru Ishikawa of Japan and these simple tools have since been widely used across industries to track quality.
RL4.2	Six-Sigma Thinking	**RL4.2 Six-Sigma Thinking** Six-Sigma is a statistical measurement and process improvement methodology pioneered by Motorola in the 80s and popularized by General Electric, USA. It believes that any process can be improved by reducing defects for opportunities. This module illustrates DMAIC—a structured methodology for reducing defects in any process.
RL4.3	Quality Culture in Japan	**RL4.3 Quality Culture in Japan** More than Process-centricity, Culture plays a vital role in institutionalization of quality in any organization. Post world-war, the success of Japanese industry was attributed to its strong emphasis on Quality embedded in its culture. This module walkthrough experiences of working with Japanese customers in development of quality software products. Also highlighted are cross-cultural differences among Indian, Japanese and Americans in their approach to Quality.
RL4.4	SQM – A Summary	**RL4.4 Software Quality Management – A Summary** This modules summarizes the concepts of Quality, Quality Assurance and Quality Management covered in this course along with the challenges faced by the Project Manager in balancing the triad of Cost, Schedule and Quality.