UNIT	TOPIC / SUB TOPIC	TEACHING HOURS
1	SYSTEM ANALYSIS AND DESIGN	10
	Software Development Models o Waterfall Model o The Incremental Model o The Spiral Model	4
	Requirement Modeling / Fact-finding techniques Interview Document review Observation Questionnaires and surveys Overview Feasibility Study Operational , Technical , Economic , Schedule Feasibility Data Flow Diagram: Concepts, Symbols, Rules, Construction of CD and DFD	6
2	**Object Oriented Analysis**	10
	The Constituents of OOAD: o Objects and Classes o Links and Association o Generalization and Specialization o Aggregation and Composition o Coupling and Cohesion o Components o Interfaces	4
	UML DIAGRAMS **Use-Case Diagram:** Benefits of Use-Case Diagram o Actors, Use-Cases , Relationship between Actor and Use Case **Sequence Diagram:** Elements of Sequence Diagram: Life Lines, Messages, Activation, Guards, Combined Fragments, Objects	6

3	**UML DIAGRAMS**	10
	Activity Diagram: Elements of Activity Diagram: Initial State, Final State Action / Activity Transitions , Decision Synchronization, Fork and Join Swim lanes, Object and Object Flow	6
	Class Diagram: Elements of Class Diagram: Class Relationship Association Class Interface and Package	4
4	**QA and TESTING**	10
	Quality, Quality Assurance and Quality Control White Box Testing Black Box Testing Integration Testing	6
	System and Acceptance Testing Performance Testing Regression Testing Test Metrics and Measurements	4

UNIT 1 :

System Analysis and Design Methods

Publisher: Cengage Learling

By: Gary B. Shelly, Thomas J. Cashman, Harry J. Rosenblatt

UNIT 2, 3

Magnifying Object-Oriented Analysis and Design

Publisher: PHI

Author: Arpita Gopal and Netra Patil

UNIT 3, 4

Software Testing: Principles and Practices

Pearson Education

Srinivasan Desikan Gopalaswamy Ramesh

UNIT 4 **QA and TESTING**

QA is process oriented.

QC is product oriented.

Testing, therefore is product oriented and thus is in the **QC** domain.

Testing for **quality** isn't assuring **quality**, it's controlling it.

Quality Assurance makes sure you are doing the right things, the right way.

UNIT 4 **QA and TESTING**

Quality assurance (**QA**) is a way of preventing mistakes and defects in manufactured products and avoiding problems when delivering solutions or services to customers; which ISO 9000 defines as "part of quality management focused on providing confidence that quality requirements will be fulfilled".

UNIT 4 **QA and TESTING**

Quality assurance (**QA**) is any systematic **process** of determining whether a product or service meets specified requirements. **QA** establishes and maintains set requirements for developing or manufacturing reliable products.

UNIT 4 **QA and TESTING**

Quality assurance standards (or **QA** for short) are a set of **standards** that have been chosen and implemented by businesses all around the world to show commitment to delivering quality products and services to customers. Specifically, **quality assurance** (or **QA**) **standards** are about meeting customer and other requirements.

UNIT 4 **QA and TESTING**

Quality : The standard of something as measured against other things of a similar kind; the degree of excellence of something.

— **Quality** is doing the right things right and is uniquely defined by each individual.

— A product or process that is Reliable, and that performs its intended function is said to be a **quality** product.

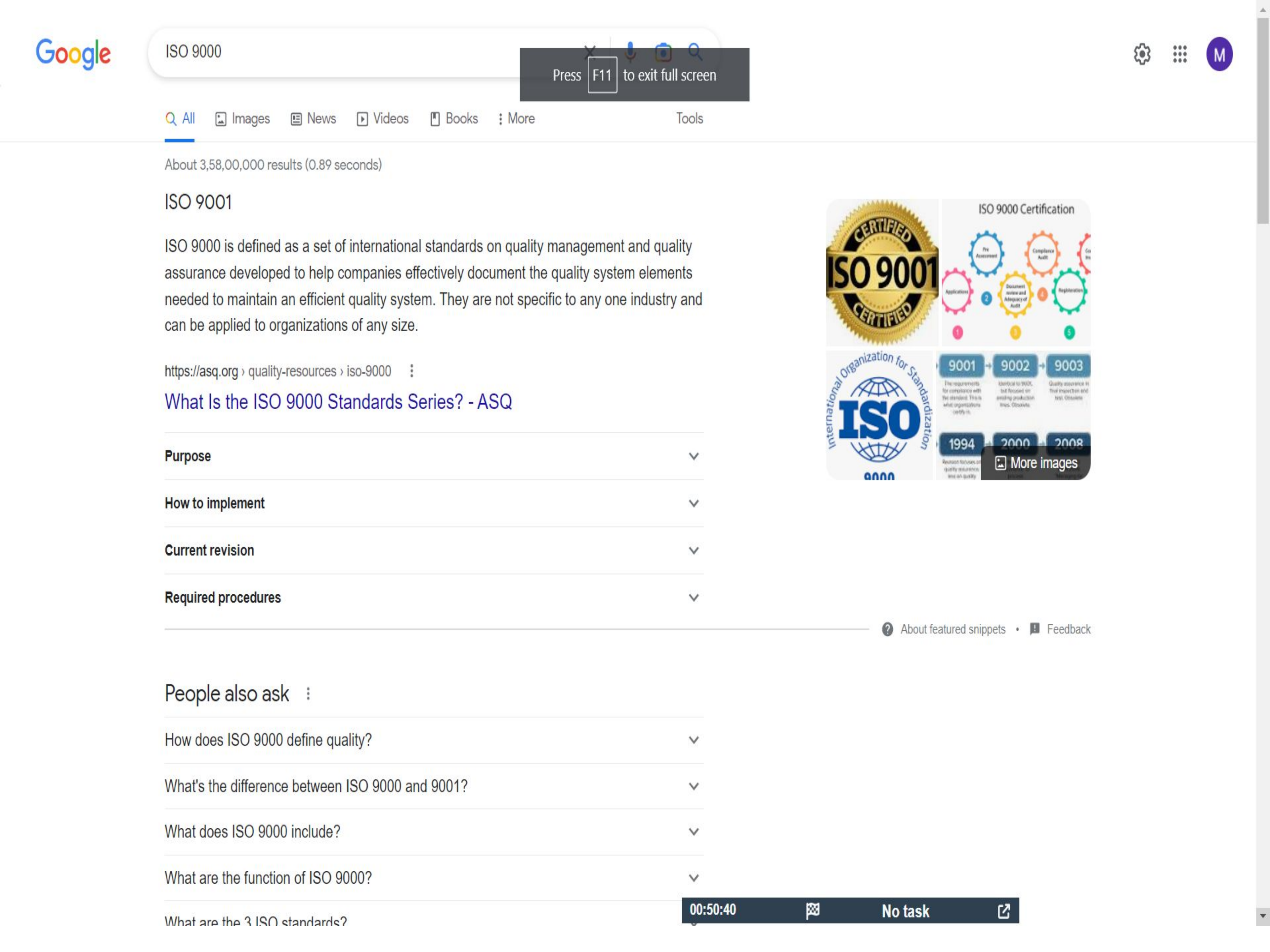
— The degree to which something meets or exceeds the expectations of its consumers.

UNIT 4 **QA and TESTING**

The **Quality Assurance** (**QA**) **role** is the **role** responsible for guaranteeing a level of quality for the end client, and to help the software development team to identify problems early in the process. It is not surprising that people in this **role** are often known as "testers". Of course, the **role** is more than just testing

UNIT 4 **QA and TESTING**

Quality assurance. **Quality assurance** (QA) is a way of preventing mistakes and defects in manufactured products and avoiding problems when delivering solutions or services to customers; which ISO 9000 defines as "part of **quality** management focused on providing confidence that **quality requirements** will be fulfilled".



About 3,58,00,000 results (0.89 seconds)

ISO 9001

ISO 9000 is defined as a set of international standards on quality management and quality assurance developed to help companies effectively document the quality system elements needed to maintain an efficient quality system. They are not specific to any one industry and can be applied to organizations of any size.

https://asq.org › quality-resources › iso-9000

What Is the ISO 9000 Standards Series? - ASQ

Purpose

How to implement

Current revision

Required procedures

People also ask

How does ISO 9000 define quality?

What's the difference between ISO 9000 and 9001?

What does ISO 9000 include?

What are the function of ISO 9000?

What are the 3 ISO standards?

UNIT 4 **QA and TESTING**

A **TEST PLAN** is a document describing software **testing** scope and activities. It is the basis for formally **testing** any software/product in a project. ISTQB Definition.**test plan**: A document describing the scope, approach, resources and schedule of intended **test** activities.

UNIT 4 **QA and TESTING**

Define

Measure

Analyze

Improve

Control

(DMAIC) model.

DMAIC is a data-driven quality strategy used to *improve* processes

UNIT 4 **QA and TESTING**

White-box testing:

White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

UNIT 4 **QA and TESTING**

White-box test design techniques include the following code coverage criteria:

Control flow testing

Data flow testing

Branch testing

Statement coverage

Decision coverage

Modified condition/decision coverage

Prime path testing

Path testing

UNIT 4 **QA and TESTING**

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

UNIT 4 **QA and TESTING**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing:

Unit testing,

integration testing,

system testing and

Acceptance. It is sometimes referred to as specification-based testing

Performance Testing

Performance specifications (requirements) are documented in a performance test plan. Ideally, this i done during the requirements development phase of any system development project, prior to an design effort.

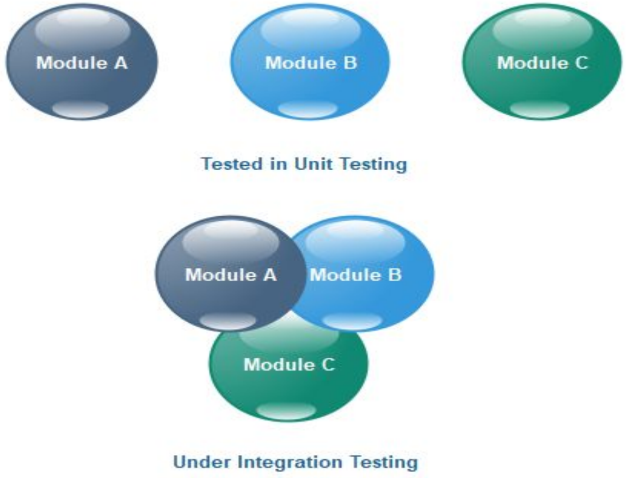
Performance specifications (requirements) should ask the following questions, at a minimum:

- In detail, what is the performance test scope? What subsystems, interfaces, components, etc. are in and out of the scope for this test?
- For the user interfaces (UIs) involved, how many concurrent users are expected for each (specif peak vs nominal)?
- What does the target system (hardware) look like (specify all server and network applianc configurations)?
- What are the time requirements for any/all backend batch processes (specify peak vs nominal)

A system along with its functional requirements must meet the quality requirements. One exampl of quality requirement is performance level. The users may have objectives for a software system in terms of memory use, response time, throughput, and delays. Thus, performance testing is to tes the run-time performance of the software on the basis of various performance factors. Performanc testing becomes important for real-time embedded systems, as they demand critical performanc requirements.

Integration testing

- Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before system testing



- **System Testing:**
  System Testing is done to check whether the software or product meets the specified requirements or not. It is done by both testers and developers. It contains the Testings: System testing, Integration Testing. It is done through more positive and negative test cases.
- Acceptance Testing**:**
  Acceptance Testing is done after the system testing. It is used to check whether the software meets the customer requirements or not. Acceptance testing is used by testers, stakeholders as well as clients. It includes only Functional Testing and it contain two testing Alpha Testing and Beta Testing.

- Regression testing
- Regression testing is re-running functional and non-functional tests to ensure that previously developed and tested software still performs as expected after a change. If not, that would be called a regression.

- **What Is Software Testing Metrics?**
- ***A Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute.***
- ***Metrics can be defined as "STANDARDS OF MEASUREMENT".***

- Software Metrics are used to measure the quality of the project. Simply, a Metric is a unit used for describing an attribute. Metric is a scale for measurement.
- Suppose, in general, "Kilogram" is a metric for measuring the attribute "Weight". Similarly, in software, "How many issues are found in a thousand lines of code?", h*ere* ***No. of issues is one measurement & No. of lines of code is another measurement. Metric is defined from these two measurements*****.**
- **Test metrics example:**
- How many defects exist within the module?
- How many test cases are executed per person?
- What is Test coverage %?

Measurement is the quantitative indication of extent, amount, dimension, capacity, or size of some attribute of a product or process.

Test Measurement example: Total number of defects.

1. How many test cases have been designed per requirement?
2. How many test cases are yet to design?
3. How many test cases are executed?
4. How many test cases are passed/failed/blocked?
5. How many test cases are not yet executed?
6. How many defects are identified & what is the severity of those defects?
7. How many test cases are failed due to one particular defect? etc.