B.B.A. (Computer Application) Semester-VI
Formerly known as B.C.A.

SOFTWARE TESTING

Dr. ARUNA A. DEOSKAR
Ms. JYOTI J. MALHOTRA
VIKAS S. TAYADE

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

A Text Book Of

SOFTWARE TESTING

For

B.B.A. (Computer Application)

Formerly Known As B.C.A. – Semester - VI

As Per Revised Syllabus Effective from June 2015

Dr. Aruna A. Deoskar
MCA, Ph.D. Computer Management
ATSS College of Business Studies
and Computer Applications (CBSCA)
Chinchwad, Pune.

Ms. Jyoti J. Malhotra
M.E. (Computer Science and Engineering)
Asst. Professor, Department of I.T.
MIT College of Engineering,
Pune

Vikas S. Tayade
M.C.A.
Asst. Professor, ATSS College of Business Studies
and Computer Applications (CBSCA)
Chinchwad, Pune.

Price ₹ 160.00

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

N3475

BCA – VI : SOFTWARE TESTING **ISBN 978-93-5164-854-3**

Third Edition : January, 2018

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312, Shivaji Nagar,
Off J.M. Road, PUNE – 411005
Tel - (020) 25512336/37/39, Fax - (020) 25511379
Email : niralipune@pragationline.com

Polyplate

Printed By :
SHIVANI PRINTERS
Shop No. 7, 8, 9 Kinara Sahakari Gruh Sanstha,
1311, Kasbapeth, Pune – 411011.
Phone (020) 24577245

DISTRIBUTION CENTRES

PUNE

Nirali Prakashan : 119, Budhwar Peth, Jogeshwari Mandir Lane, Pune 411002, Maharashtra
Tel : (020) 2445 2044, 66022708, Fax : (020) 2445 1538
Email : bookorder@pragationline.com, niralilocal@pragationline.com

Nirali Prakashan : S. No. 28/27, Dhyari, Near Pari Company, Pune 411041
Tel : (020) 24690204 Fax : (020) 24690316
Email : dhyari@pragationline.com, bookorder@pragationline.com

MUMBAI

Nirali Prakashan : 385, S.V.P. Road, Rasdhara Co-op. Hsg. Society Ltd.,
Girgaum, Mumbai 400004, Maharashtra
Tel : (022) 2385 6339 / 2386 9976, Fax : (022) 2386 9976
Email : niralimumbai@pragationline.com

DISTRIBUTION BRANCHES

JALGAON

Nirali Prakashan : 34, V. V. Golani Market, Navi Peth, Jalgaon 425001,
Maharashtra, Tel : (0257) 222 0395, Mob : 94234 91860

KOLHAPUR

Nirali Prakashan : New Mahadvar Road, Kedar Plaza, 1st Floor Opp. IDBI Bank
Kolhapur 416 012, Maharashtra. Mob : 9850046155

NAGPUR

Pratibha Book Distributors : Above Maratha Mandir, Shop No. 3, First Floor,
Rani Jhanshi Square, Sitabuldi, Nagpur 440012, Maharashtra
Tel : (0712) 254 7129

DELHI

Nirali Prakashan : 4593/21, Basement, Aggarwal Lane 15, Ansari Road, Daryaganj
Near Times of India Building, New Delhi 110002, Mob : 08505972553

BENGALURU

Pragati Book House : House No. 1, Sanjeevappa Lane, Avenue Road Cross,
Opp. Rice Church, Bengaluru – 560002.
Tel : (080) 64513344, 64513355,Mob : 9880582331, 9845021552
Email:bharatsavla@yahoo.com

CHENNAI

Pragati Books : 9/1, Montieth Road, Behind Taas Mahal, Egmore,
Chennai 600008 Tamil Nadu, Tel : (044) 6518 3535,
Mob : 94440 01782 / 98450 21552 / 98805 82331,
Email : bharatsavla@yahoo.com

niralipune@pragationline.com | www.pragationline.com

Also find us on www.facebook.com/niralibooks

Preface ...

We take this opportunity to present this book entitled as **"Software Testing"** to the students of Sixth Semester of VI BBA. (Computer Application). The objective of this book is to present the subject matter in a most concise and simple manner. The book is written strictly according to the Revised Syllabus of the University.

The book has its own unique features. It brings out the subject in a very simple and lucid manner for easy and comprehensive understanding of the basic concepts, its intricacies, procedures and practices. This book will help the readers to have a broader view on Software-testing Concepts. The language used in this book is easy and will help students to improve their vocabulary of Technical terms and understand the matter in a better and easier way.

We sincerely thank Shri. Dineshbhai Furia and Shri. Jignesh Furia of Nirali Prakashan, for the confidence shown with us and giving us this opportunity to reach out to the students of BCA.

We thanks all the contributors who have helped us directly or indirectly to compile this book and bring it in the market.

We thank Mrs. Anita Panajkar and Mrs. Aabha Athavale for their important inputs time to time and Mr. Santosh Bare, Mrs. Deepa Sawant who painstakingly attended to all the details to make this book appear good.

We also thank Mr. Ravindra Walodare, Mr. Sachin Shinde, Nikunj Joshi, Nilesh Deshmukh, Ashok Bodke, Moshin Sayyed and Nitin Thorat.

We have given our best inputs for this book. Any suggestions towards the improvement of this book and sincere comments are most welcome on niralipune@pragationline.com.

AUTHORS

Syllabus ...

1. SOFTWARE TESTING (6 Lectures)

Introduction, Nature of errors, Testing principles and Testing fundamentals, Debugging.

2. APPROACHES TO TESTING – I (10 Lectures)

White Box Testing, Black Box Testing, Gray Box Testing, Unit Testing.

Integration – Top down, Bottom up, Big-bang, Sandwich.

3. TESTING FOR SPECIALIZED ENVIRONMENTS (10 Lectures)

Testing GUI's, Testing of Client/Server Architectures, Testing Documentation and Help facilities, Testing for Real-Time systems.

4. SOFTWARE TESTING STRATEGIES AND SOFTWARE METRICS (12 Lectures)

Validation Testing, System Testing, Verification, Performance Testing, Regression Testing, Agile Testing, Acceptance Testing, Smoke Testing, Load Testing, Introduction, Basic Metrics, Complexity Metrics.

5. SPECIALIZED TESTING AND TESTING TOOLS (INTRODUCTION) (6 Lectures)

Test case design, JUnit, Apache JMeter, Winrunner, Loadrunner, Rational Robot.

•••

Contents ...

•••

Chapter 1...

Software Testing

Contents ...

1.1 INTRODUCTION TO SOFTWARE TESTING (Oct. 16, April 16)

Definition :

"Software testing is a process used to identify the correctness, completeness & quality of developed software"

Testing is the process of exercising and evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements.

Primary role of testing is not demonstration of correct performance, but the exposure of hidden defects. - Glen Myers

Testing is concerned with *errors, faults, failures and incidents*. A test is the act of exercising software with an objective of

- Finding failure
- Demonstrate correct execution - Paul Jorgensen

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Software testing is a process used to identify the correctness, completeness, and quality of developed computer software**.** It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end users.

OR

In simple words, software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free.

OR

It can also be stated as the process of validating and verifying that a software program or application or product:

- Meets the business and technical requirements that guided it's design and development
- Works as expected; and
- Can be implemented with the desired characteristics.

Testing Objectives **(April 17)**

- Testing is a process of executing a program with the intention of finding Errors.
- Establishing confidence that a program does what it is supposed to do. It is the measurement of Software Quality which confirms that a program performs its intended functions correctly.
- Testing identify difference between Expected and Actual Result. This is a process of trying to discover every conceivable fault or weakness in a work product.

Benefits of Software Testing :

- Primary benefit of testing is - It improves the Quality as testing looks for a class of problems and flags them as possible candidates for investigation and fixes.
- Secondary benefit is - Testing demonstrates that software functions appear to be working according to specifications.

Goals of Software Testing

- The main goal of software testing is to find bugs as early as possible and to fix bugs and to make sure that the software is bug free.
- The goals of software testing may be classified in to three major categories as shown in Fig. 1.1.

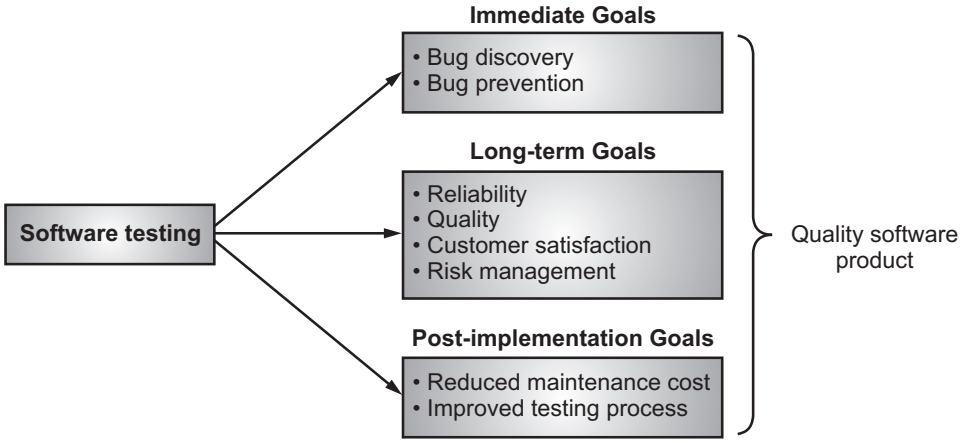

Fig. 1.1 : Software testing goals

1. **Immediate Goals:** These goals are also called as short term goals. These testing goals are the immediate result after testing. These goals contain:
   - **(i) Bug discovery:** Is the immediate goal of software testing to find errors at any stage of software development. Number of the bugs are discovered in early stage of testing.
   - **(ii) Bug prevention:** Is the consequent action of bug discovery.
2. **Long Term Goals:** These testing goals affect the software product quality in the long term. These include:
   - **(i) Quality:** This goal enhances quality of the software product.
   - **(ii) Customer satisfaction:** This goal verifies the customers satisfaction for developed software product.
   - **(iii) Reliability:** It is a matter of confidence that the software will not fail. In short reliability means to gain the confidence of the customers by providing them a quality product.
   - **(iv) Risk management:** Risk must be done to reduce the failure of product and to manage risk in different situations.
3. **Post Implemented Goals:** These goals are important after the software product released. Some of them are listed below:
   - **(i) Reduce maintenance cost:** Post-released errors are costlier to fix and difficult to identify.
   - **(ii) Improved software testing process:** These goals improve the testing process for future use of software projects. These goals are known as post-implementation goals.

1.2 NATURE OF ERRORS **(April 16)**

Software Bug

A software bug is the common term used to describe an error, flaw, mistake, failure or fault in a computer program or system that produces an incorrect or unexpected result, or which causes the program to perform in an un-intended manner.

Most bugs arise from mistakes and errors in either a program's source code or its design, but the main cause can be traced to the specification.

A Software bug occurs when one or more of following is true :

- The Software doesn't do something that the product specification says it should do.
- The Software does something that the product specification says it should not do.

Errors

The system is in a state such that further processing by the system will lead to a failure.

An error is a **mistake**, **misconception** or **misunderstanding** on the part of a software developer. It might be typographical error, a misleading of specifications, a misunderstanding of what a subroutine does.

The errors may be :

- Actual bugs in the code.

- Incorrect implementation of the requirements or functional specifications because of Misunderstandings and/or Incomplete requirements or functional specifications.
- Incorrect human action that produces erroneous step, process, or inaccurate result.
  User interface errors i.e. Functionality, Communication, Command structure, Missing commands ,Performance, Output
- Calculation errors.
- Errors in handling or interpreting data.
- Documentation - The user does not observe operation described in manuals.
- Testing errors.

Faults

A fault occurs when a human error results in a mistake in some software product(s).

For example, a developer might misunderstand a user-interface requirement and therefore create a design that includes misunderstanding. The design fault can also result in incorrect code, as well as incorrect instructions in user manual.

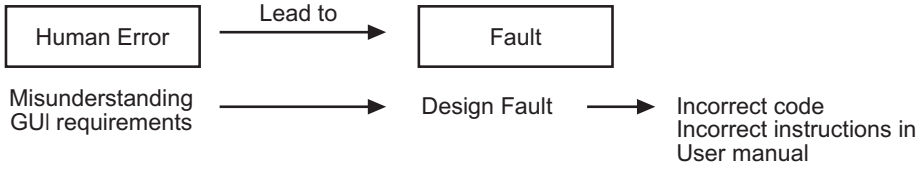

Fig. 1.2 : Flow of Fault

A fault is the difference between incorrect program and the correct version. A single error can result in one or more faults. One fault can result in multiple changes to one product (such as changing several sections of a piece of code) or multiple changes to multiple products (such as change to requirements, design code and test plans etc).

Examples of Faults and Errors

Faults in the Interface specification

- Mismatch between what the client needs and what the server offers.
- Mismatch between requirements and implementation.

Algorithmic Faults

- Missing initialization.
- Branching errors (too soon, too late).

Mechanical Faults (very hard to find)

- Documentation does not match actual conditions or operating procedures.

Errors

- Stress or overload errors.
- Capacity or boundary errors.
- Timing errors.
- Throughput or performance errors.

A fault (bug) can go undetected until a failure occurs, which is when a user or tester perceives that the system is not delivering the expected service.

Failure

Failures can be defined as :

- Deviation of the system from its expected behaviour or service.
- The inability of a system or component to perform its required functions within specified performance requirements.

Failures can be discovered both before and after system delivery, as they can occur in testing as well as in operation.

Faults represent problems that the developer sees, while failures are problems that the user sees.

Ideally the life of a bug ends when it is uncovered in testing and fixed.

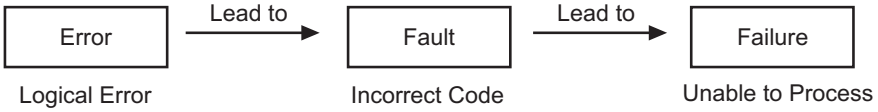

Fig. 1.3 : Flow of Failure

For example : Consider ATM machine

Developer might have forgotten to fire update query on Database when user changes the PIN number, and screen displays the message "Your PIN number has been changed"!! When user tries to access the system with new PIN number….. System gives the message "Sorry unable to process".

Defects **(April 17)**

Defects are :

- Abnormal behaviour of the software.
- Non-conformance to requirements or functional/program specification.
- Errors in Testing.
- Mistakes in Correction.

For example : Requirements and specification defects, design defects, coding defects or testing defects.

Software Quality

Software quality is the degree of conformance to explicit or implicit requirements and expectations of user/customer.

This is a field of study and practice that describes the desirable attributes of software products. Though a fixed software quality model is often helpful for considering an overall understanding of software quality, in practice the relative importance of particular software characteristics typically depends on software domain, product type and intended usage.

Thus, software characteristics should be defined for, and used to guide the development of, each product.

Quality function deployment provides a process for developing products based on characteristics derived from user needs.

Definition:

Software quality can be defined as "an effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it".

OR

It can also be defined as "the degree to which a system, component, or process meets specified requirements and the degree to which a system, component, or process meets customer or user needs or expectations".

OR

Software quality can be defined as, "the totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs".

OR

According to Roger Pressman Software Quality can be defined as "conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software".

Quality Assurance(QA) :

This is a process driven approach, it is a process to monitor and improve existing quality processes. It is a process of verifying whether the software product or services meets or exceeds the customer expectations. It ensures that the product or services are developed or implemented on agreed standards.

Quality assurance ensures that the processes designed for the product development and services are effective enough to meet the objectives. It prevents the software defects/errors.

Quality Control (QC) :

The goal of a software tester is to find bugs, find them as early as possible, and make sure they get fixed. A software quality assurance person's main responsibility is to create and enforce standards and methods to improve the development process and to prevent bugs from ever occurring. Quality control is a process used to ensure quality in a product or a service.

Quality Control (QC) is the observation techniques and activities used to fulfill requirements for quality.Quality control activities are work product oriented. They measure the product identity deficiencies and suggests improvements.

Quality Control (QC) can be defined as "the processes and methods used to monitor work and observe whether requirements are met".

It includes the following activities:

1. **Reviews:**
   (i) Requirement Review
   (ii) Design Review
   (iii) Code Review
   (iv) Deployment Plan Review
   (v) Test Plan Review
   (vi) Test Cases Review
2. **Testing:**
   (i) Unit Testing
   (ii) Integration Testing
   (iii) System Testing
   (iv) Acceptance Testing

People may get confused about the differences among Quality Assurance, Quality Control, and Testing. Although they are interrelated and to some extent, they can be considered as same activities, but there exist distinguishing points that set them distant.

The following table lists the points that differentiate QA, QC, and Testing.

Quality Assurance	Quality Control	Testing
QA includes activities that make sure the implementation of processes, procedures and standards are in context to verification and validation of developed software and intended requirements.	It includes activities that make sure the verification and validation of a developed software with respect to documented (or not in some cases) requirements.	It includes activities that make sure the identification of bugs/error/defects in a software, statically and dynamically.
Focuses on processes and procedures rather than conducting actual testing on the system.	Focuses on controls by executing the software with an aim to identify bug/defect through implementation of procedures and process.	Focuses on actual testing.
Process-oriented activities.	Product-oriented activities.	Product-oriented activities.
Preventive activities.	It is a preventive and corrective process.	It is a preventive and corrective process.
It is a subset of quality management system.	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

Role of Testing :

It is good to begin testing from the first stage, to avoid complexity by fixing the errors at the last stage.

We do software testing for many reasons, such as :

1. To check the reliability of the software.
2. To be ensured that the software does not contain any bug which can become a reason for failure?
3. To check the software was made according to its specification.
4. To check that the software meets its requirements.
5. To check that users are capable of using the software.
6. To check software works with other software and hardware it needs to work with.
7. To improve quality of software by removing maximum possible errors or defects from it.

In any organization there are Roles that must be fulfilled within any testing organization. The requirement for any given role depends on the size, complexity, goals, and maturity of the testing organization.

It is quite possible that one person could fulfill many roles within the testing organization. Some testing roles are given below:

1. **Test Lead or Test Manager:** The Role of Test Lead / Manager is to effectively lead the testing team. To fulfill this role the Lead must understand the discipline of testing and how to implement a testing process effectively while fulfilling roles of a manager. That means the manager must manage and implement or maintain an effective testing process.
2. **Test Architect:** The Role of the Test Architect is to formulate an integrated test architecture that supports the testing process and leverages the available testing infrastructure. To fulfill this role, the Test Architect must have a clear understanding of the short-term and long-term goals of the organization, the resources (both hard and soft) available to the organization and a clear vision on how to most effectively deploy these assets to form integrated test architecture.
3. **Test Designer or Tester:** The Role of the Test Designer / Tester is to design and document test cases, execute tests, record test results, document defects, and perform test coverage analysis. To fulfill this role the designer must be in position to apply the most appropriate testing techniques to test the application as efficiently as possible while meeting the test organizations testing mandate.
4. **Test Automation Engineer:** The Role of the Test Automation Engineer is to create automated test case scripts that perform the tests as designed by the Test Designer. To fulfill this role the Test Automation Engineer must develop and maintain an effective test automation infrastructure using the tools and techniques available to the testing organization. The Test Automation Engineer must work in concert with the Test Designer to ensure the appropriate automation solution is being deployed.

Software Testing

Publisher : Nirali Prakashan

ISBN : 9789351648543

Author : Dr. Aruna A. Deoskar, Ms. Jyoti J. Malhotra, Vikas S. Tayade