Software Testing Handbook

My Testing Days of Modern Arena

Dr. Arupratan Santra

Himalaya Publishing House
ISO 9001:2015 CERTIFIED

Software Testing Handbook

My Testing Days of Modern Arena

Dr. Arupratan Santra

Himalaya Publishing House
ISO 9001:2015 CERTIFIED

First Edition : 2019

Published by : Mrs. Meena Pandey for **Himalaya Publishing House Pvt. Ltd.,**
"Ramdoot", Dr. Bhalerao Marg, Girgaon, Mumbai - 400 004.
Phone: 022-23860170, 23863863, **Fax:** 022-23877178
E-mail: himpub@vsnl.com; **Website:** www.himpub.com

Branch Offices :

New Delhi : "Pooja Apartments", 4-B, Murari Lal Street, Ansari Road,
Darya Ganj, New Delhi - 110 002.
Phone: 011-23270392, 23278631; Fax: 011-23256286

Nagpur : Kundanlal Chandak Industrial Estate, Ghat Road, Nagpur - 440 018.
Phone: 0712-2738731, 3296733; Telefax: 0712-2721216

Bengaluru : Plot No. 91-33, 2nd Main Road, Seshadripuram,
Behind Nataraja Theatre, Bengaluru - 560 020.
Phone: 080-41138821; Mobile: 09379847017, 09379847005

Hyderabad : No. 3-4-184, Lingampally, Besides Raghavendra Swamy Matham,
Kachiguda, Hyderabad - 500 027.
Phone: 040-27560041, 27550139

Chennai : New No. 48/2, Old No. 28/2, Ground Floor, Sarangapani Street,
T. Nagar, Chennai - 600 012. Mobile: 09380460419

Pune : "Laksha" Apartment, First Floor, No. 527, Mehunpura,
Shaniwarpeth (Near Prabhat Theatre), Pune - 411 030.
Phone: 020-24496323, 24496333; Mobile: 09370579333

Lucknow : House No. 731, Shekhupura Colony, Near B.D. Convent School,
Aliganj, Lucknow - 226 022.
Phone: 0522-4012353; Mobile: 09307501549

Ahmedabad : 114, "SHAIL", 1st Floor, Opp. Madhu Sudan House, C.G. Road,
Navrang Pura, Ahmedabad - 380 009.
Phone: 079-26560126; Mobile: 09377088847

Ernakulam : 39/176 (New No. 60/251), 1st Floor, Karikkamuri Road, Ernakulam,
Kochi - 682 011. Phone: 0484-2378012, 2378016;
Mobile: 09387122121

Bhubaneswar : Plot No. 214/1342, Budheswari Colony, Behind Durga Mandap,
Bhubaneswar - 751 006. Phone: 0674-2575129;
Mobile: 09338746007

Kolkata : 108/4, Beliaghata Main Road, Near ID Hospital, Opp. SBI Bank,
Kolkata - 700 010. Phone: 033-32449649; Mobile: 07439040301

DTP by : Sunanda

Printed by : M/s. Aditya Offset Process (I) Pvt. Ltd., Hyderabad. On behalf of HPH.

Dedicated to

My Parents Mr. H.N. Santra and Mrs. B. Santra

&

Beloved wife Mrs. Anindita Das and daughter Arunima Santra

Introduction

The demand or usage of software is increasing day by day globally at a rapid pace. This growth of software usage needs to have error-free application as nowadays these are used in various workspaces in human life. Any single error in these applications will cause huge loss in revenue or human life. These errors are rectified with continuous focus on software testing, which will help to build high-quality software for any organisation. This is extremely important for an organisation that seeks high-quality product delivery to its client to maintain high reputation in the market.

Software testing methods are used to rectify error in any software from day one when software development started, way back in 1950. This method also needs improvement in process, analysis, methodology, approach, etc. Software testing is one of the key steps in software development life cycle (SDLC). Testing approach is evolving year after year due to various types of software applications used in society.

Due to recent trend in SDLC: V-model, DevOps or Agile, software testing approach is upgraded, rather it has evolved in a new dimension. This evolution has been linked with specific methods or testing approach based software usage and complexity. These approaches are different than traditional approach for strengthening the product quality. Software testing follows either manual or automation or both approaches, a framework to minimise the defects.

Testing framework is natural selection process based on feasibility, but not a prescriptive approach. Test team can choose testing framework in different ways, relating to test team capability, analytical ability, organisation goal, etc. This book describes the modern aspects or approach of software testing in different conditions.

Rather than just explaining the various traditional testing approach and methodology, this book will help modern age testing team to implement new testing approaches or will allow us to think of a new approach for each software development. These approaches can help practitioner, manager, architect or beginner. This book also describes about test metrics, test effort estimation and various aspects of modern testing approach for recent trendy application, e.g., IoT, RPA, etc. This book includes more than ten chapters. It begins with history of software testing followed by defect predictability and software test effort estimation. It also describes about various automation testing approaches followed by IoT and RPA testing. It is not intended to provide extensive literature survey or detailed explanation of test approach or test design in this book. The main goal is to describe the modern aspects of testing which are consolidated in this book collectively under one umbrella. It does not suggest that these approaches are only best or optimal for modern aspect of software testing or does not suggest to use only these approaches – judgement is left to the readers (test engineer, project manager, test manager, scrum master, students, etc.). This book has various case studies with examples and many algorithms. It is possible to say that it is the author's 17+ years of experience in software testing and test program management. Any suggestion or comments can be sent to arsantra@rediffmail.com about this book.

Anindita Das

Scientist, GoI

Acknowledgements

Firstly, I would like to thank all my colleagues of Cyient and Infosys as they are my inspiration, they helped and motivated me to complete my book. I thank my senior management (Prem Kumar J., Dr. D. Rao, Avadhanulu, Venkata D., Shishank G., Niranjan, Archana Das and my other family members) who always motivated me to write many publications.

Many others have contributed to make my dream a reality for publishing this book. My sincere thanks to my all co-authors (mentioned in the reference section) who helped me to depict various testing concepts in various national or international journals, conferences or magazines.

I am particularly thankful to my wife who always encouraged me for the completion of this project, and my daughter Arunima, who had to sacrifice their legitimate portion of time. I thank Sanjeev K. Balodia, who provided enormous help in completing the material, diagram, etc.

Author

Contents

1 Software Testing – State of Art Review

Software testing is a major focused step of software development life cycle (SDLC) for quality assessment in Software Engineering. The basic principle of testing for correctness is the selection of test cases that satisfy some particular criterion.

Software testing is the activity of establishing whether a program correctly implements a given model. Testing techniques can be distinguished among static and dynamic analysis techniques. Static analysis techniques do not require the program under test to be executed, while dynamic analysis techniques do. Example of static analysis techniques are formal proofs of correctness, code inspections, data flow analysis. Examples of dynamic techniques are testing techniques. A concise history of software testing as per Gelperin & Hetzel till 1996 and beyond, is summarised as below:

Table 1.1: Evolution of Testing

-1956	The debugging oriented period
1957-1978	The demonstration oriented period
1979-1982	The destruction oriented period
1983-1987	The evaluation oriented period
1988-1996	The prevention oriented period
1997-2005	The reduced Cost of Quality (CoQ) period
2006-2010	The platform oriented QA period
2011-2015	The hybrid competitive oriented QA period
2016-2020	Agile/DevOps and Cloud Oriented QA period
2021	IOT & RPA based application QA period

Debugging Oriented Testing

Programs are written, and then simply ‘checked out’ by the programmers until they are satisfied that all the bugs have been identified and removed. Some or all of the identification and removal activities are described as ‘testing’; there is no real consensus as to what the ‘testing’ part is.

The criteria used for selecting test cases are entirely ad hoc, and based on the programmers’ experience and their understanding of the system being built.

Demonstration Oriented Testing

Debugging and testing are identified as separate activities.

> ***Debugging*** (or ‘sanity’ testing) consists of ensuring that the system runs (i.e., doesn’t crash).
>
> ***Testing*** consists of ensuring that the system does what it is supposed to.

Testing is performed with the aim of showing that the system conforms to its requirements.

Destruction Oriented Testing

Testing gained acceptance from Myers’ book ‘*The Art of Software Testing*’, which described testing as:

> “The process of executing a program with the intention of finding errors”.

The meanings of the words testing and debugging are different again:

> ***Testing*** is concerned with revealing the presence of faults in the system.
>
> ***Debugging*** is concerned with locating and correcting those faults.

The criterion for test selection chooses test cases that reveal particular faults if they are present in the system.

These last two periods separate testing out from other parts of software development, as a distinct and final stage in the life cycle model. The emphasis is on the actual execution of the test cases by the implementation of the system.

Evaluation Oriented Testing

Along with other analysis and review techniques, testing is integrated into an evaluation phase at the end of every stage of life cycle. This period began with the publication in 1983 by the National Bureau of Standards (USA), and the realisation that the earlier in the life cycle a fault is detected, the less costly it is to correct.

For every stage in the life cycle there are requirements and products. The collective goal of the evaluation phase is to attempt to measure how well the products meet their requirements.

Prevention Oriented Testing

This approach to testing was initiated by Hetzel and Gelperin, who generalised methods for unit testing, and developed a comprehensive methodology for practical test management.

The philosophy is to prevent errors in each stage of the life cycle model by using testing and other evaluation techniques as the stage progresses. (cf. the evaluation period, where testing and other evaluation techniques were only used at the end of each life cycle phase.) The criterion is now directed at finding places where errors might be made.

The Reduced CoQ Period

This concept has come to testing industry with the releases of various automated testing tools, e.g., Winrunner in 1997, Astra Quick Test in Feb, 2001 and other few automated testing tools. Test Maturity Model (TMMi) process has started in same time (1996 onwards). Matured software industry always looks for reduced cost software development. As an initiative, the industry started using concept of reduction in CoQ.

> Experts define it as “the process of executing a program with the intention of finding errors with faster path and reduced elapsed time”.

Test automation helps to execute the test scripts 20-30 per cent faster and detects similar kinds of defects. TMMi is used to establish lean testing process to reduce testing and review cost – ultimately CoQ reduced gradually.

The Platform Oriented QA

After using automated reduced cost testing services, industry opted for specialised testing services based on various platform based tools and operations.

> Experts define it as “the process of executing a platform level program with the intention of finding errors with easy and internal code platform in reduced time. Thereafter, automated testing tools are used for end to end testing”.

Testing of service virtualisation platform is done by SOAPUI tool since October 2005. Later Zimki, the world’s first platform as service (PaaS) 2005 is used for PaaS testing. Application developed in Microsoft platform are tested preferably by visual studio test suite (VSTS). In 2007 – smartphone platform has come into practice. Industry uses customised testing practices to detect defects or faults. There are various domain platforms e-commerce/GIS/financial/embedded platform, etc.

The Hybrid Competitive Oriented QA Period

Debugging and each stages of testing are separate activities:

> ***Debugging*** (or ‘E2E Regression, unit’ testing) ensures that system runs (i.e., doesn’t crash).
>
> ***Testing*** consists of ensuring of hybrid of steps that the system does what it is supposed to.

Testing happens in each and every step of the program. Regression testing happens locally using various tools/platforms. Later, test results are integrated to get the final result. This hybrid approach is used to reduce CoQ by faster schedule, low production defects by identifying more system testing defects,

Agile/DevOps and Cloud Oriented Application QA Period

Testing ‘Cloud’ applications should be done as though you would test any existing web application with a few added test cases to test the additional ‘Cloud’ features. If you follow any previous approach, then all defects will be surfaced out. Special care has to be taken for dynamic scaling, device synchronisation and automated provisioning.

Agile/DevOps testing also follows standard testing life cycle as previous categories. This is a small sprint based testing approach which helps to identify defects/faults for each sprint. So, test team will not be held till last release of delivery for testing. The major challenges are in this period is automation of Agile/DevOps based developed application. Behavioural Development and Deployment (BDD) based automation or model based (Tosca, 2016) automations are used for faster quality products.

IOT and RPA Based Application QA Period

In the age of smart city/smart home concepts across world, there will be revolution in the software industry. Applications are used through internet/cloud and they will access the various appliances through sensors/RFID, etc. Testing of these applications are needed to detect errors and faults to minimise human loss/damages.

RPA application will be used to avoid human intervention for repetitive jobs. Testing of these applications will be around 65 per cent by 2030.