

## Improving the Software Quality by Designing an Effective Test Cases with an Unit Testing

Nirmal Kumar A<sup>1</sup>, Dr. B.G. Geetha<sup>2</sup>

<sup>1</sup> Assistant Professor, Department of Computer Science and Engineering, Christian College of Engineering and Technology, Dindigul, Tamilnadu - 624619, India.

<sup>2</sup> Professor & Head, Department of Computer Science and Engineering, KSR College of Technology, Thiruchengode, Tamilnadu - 637215, India.

Corresponding Author Email: [nirmal15483@gmail.com](mailto:nirmal15483@gmail.com)

**Abstract:** Software projects are acting important role in human life and they are becoming most challenging for the software programmers. Human cannot do anything in this world without software in our day-to-day life. Software companies are developing the software products in the domains like Health Care, Financial Management, Banking, Insurance, Retail Management, etc. New methodologies, new techniques, new tools and new standards are increased day-by-day to ensure the high quality software. Since there is a stress for software employees to produce the high quality software. Both the programmers and the testers of the software should focus on the quality matters. Only the effective software testing can ensure the quality software products and satisfying the user's requirements, needs and expectations. The software testing contains several steps. Test Case design is one of the steps in Testing Life Cycle (TLC). The testers should concentrate on this test case design process carefully. In this research, the Traditional Software (Financial Management, Banking, Insurance, Retail Management) is taken as example and discussed about the improving the quality of the software.

[Nirmal Kumar A, Geetha B G. **Improving the Software Quality by Designing an Effective Test Cases with an Unit Testing** *Life Sci J* 2013;10(7s):608-610] (ISSN:1097-8135). <http://www.lifesciencesite.com>. 97

**Keywords:** Software Quality, Software Testing, Test Cases, Unit Testing

### 1. Introduction

An airline reservation system (ARS) is part of the so-called passenger service systems (PSS), which are applications supporting the direct contact with the passenger. The airline reservations system was one of the earliest changes to improve efficiency. ARS eventually evolved into the computer reservations system (CRS). A computer reservation system is used for the reservations of a particular airline and interfaces with a global distribution system (GDS) which supports travel agencies and other distribution channels in making reservations for most major airlines in a single system. Airline reservations systems contain airline schedules, fare tariffs, passenger reservations and ticket records. An airline's direct distribution works within their own reservation system, as well as pushing out information to the GDS. A second type of direct distribution channel is consumers who use the internet or mobile applications to make their own reservations [2]. Travel agencies and other indirect distribution channels access the same GDS as those accessed by the airlines' reservation systems, and all messaging is transmitted by a standardized messaging system that functions on two types of messaging that transmit on SITA's HLN [high level network]. These message types are called Type A [usually EDIFACT format] for real time interactive communication and Type B [TTY] for informational and booking type of messages. Message construction standards set by

IATA and ICAO, are global, and apply to more than air transportation. Since airline reservation systems are business critical applications, and their functionally quite complex, the operation of an in-house airline reservation system is relatively expensive. [6].

As of February 2009, there are only three major GDS providers in the market space: Amadeus, Travelport (which operates the Apollo, Worldspan and Galileo systems), Sabre and Shares. There is one major Regional GDS, Abacus, serving the Asian marketplace and a number of regional players serving single countries, including Travelsky (China), Infini and Axxess (both Japan) and Topas (South Korea). Of these, Infini is hosted within the Sabre complex, Axxess is in the process of moving into a partition within the Worldspan complex, and Topas agencies will be migrating into Amadeus [11].

An airline's inventory contains all flights with their available seats. The inventory of an airline is generally divided into service classes (e.g. first, business or economy class) and up to 26 booking classes, for which different prices and booking conditions apply. Inventory data is imported and maintained through a schedule distribution system over standardized interfaces. One of the core functions of the inventory management is the inventory control. Inventory control steers how many seats are available in the different booking classes, by opening and closing individual booking classes for

sale. In combination with the fares and booking conditions stored in the Fare Quote System the price for each sold seat is determined. In most cases inventory control has a real time interface to an airline's Yield management system to support a permanent optimization of the offered booking classes in response to changes in demand or pricing strategies of a competitor.

## 2. Material and Methods

Users access an airline's inventory through an availability display. It contains all offered flights for a particular city-pair with their available seats in the different booking classes. This display contains flights which are operated by the airline itself as well as code share flights which are operated in co-operation with another airline. If the city pair is not one on which the airline offers service it may display a connection using its own flights or display the flights of other airlines. The availability of seats of other airlines is updated through standard industry interfaces. Depending on the type of co-operation it supports access to the last seat (last seat availability) in real-time. Reservations for individual passengers or groups are stored in a so-called passenger name record (PNR). Among other data, the PNR contains personal information such as name, contact information or special services requests (SSRs) e.g. for a vegetarian meal, as well as the flights (segments) and issued tickets. Some reservation systems also allow to store customer data in profiles to avoid data re-entry each time a new reservation is made for a known passenger. In addition most systems have interfaces to CRM systems or customer loyalty applications (aka frequent traveller systems). Before a flight departs the so-called passenger name list (PNL) is handed over to the departure control system that is used to check-in passengers and baggage. Reservation data such as the number of booked passengers and special service requests is also transferred to flight operations systems, crew management and catering systems. Once a flight has departed the reservation system is updated with a list of the checked-in passengers (e.g. passengers who had a reservation but did not check in (no shows) and passengers who checked in, but didn't have a reservation (go shows)). Finally data needed for revenue accounting and reporting is handed over to administrative systems [1] [7].

Software testing is an investigation conducted to provide stakeholders with information about the

quality of the product or service under test.[1] Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects). [10].

Software testing, depending on the testing method employed, can be implemented at any time in the software development process. Traditionally most of the test effort occurs after the requirements have been defined and the coding process has been completed, but in the Agile approaches most of the test effort is on-going. As such, the methodology of the test is governed by the chosen software development methodology. [12] [3] [8]. Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Rather than replace traditional QA focuses, it augments it. Unit testing aims to eliminate construction errors before code is promoted to QA; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development and QA process. [1].

## 3. Results

Many programming groups are relying more and more on automated testing, especially groups that use test-driven development. There are many frameworks to write tests in, and continuous integration software will run tests automatically every time code is checked into a version control system. While automation cannot reproduce everything that a human can do (and all the ways they think of doing it), it can be very useful for regression testing. However, it does require a well-developed test suite of testing scripts in order to be truly useful [4] [9].Figure:2 explained about the testing the software.

**American Airlines** Home | Login | My Account | English | Contact AA | FAQ Search AA.com ... GO

Reservations - Travel Information - Travel Deals - AAdvantage® - Products & Gifts - Business Programs - About Us -

Find Flights **Choose Flights** Travelers Trip Options Review & Pay Finish

2. Choose Flights

Select a flight number to view Flight Performance Information. Search alternate dates?

Departure Thu Nov 17 from \$160

Fares per person, in US Dollars (USD), additional taxes and fees will apply.

Sort By: # of Stops	Departure	Arrival	Economy Super Saver	Economy Saver	Economy Flexible	Instant Upgrade	First Social	First Flexible
AA 1976	09:00 AM DFW	12:35 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 2525	07:25 AM DFW	11:10 AM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 1148	11:15 AM DFW	02:55 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 645	05:45 AM DFW	09:20 AM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 651	12:30 PM DFW	04:10 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 1248	02:15 PM DFW	05:55 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 484	04:15 PM DFW	07:55 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 1500	05:05 PM DFW	08:45 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 2446	07:10 PM DFW	10:45 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514
AA 1688	08:15 PM DFW	11:50 PM MIA	\$272	\$445	\$1090	\$949	Not Available	\$1514

Page 1 from \$272 | Page 2 from \$160 | Page 3 from \$160 | Page 4 from \$160 | View All Results

Optional Login

**AAdvantage members:** To continue the booking process using your saved preferences, please login below.

Your password is case sensitive and must be 6-12 numbers and/or letters.

AAdvantage Number:

Password:

Remember my AAdvantage Number on this computer.

**Continue Without Login:** If you do not wish to Login or you do not have an AAdvantage Membership.

OR

Not an AAdvantage Member? Start earning your miles today. Enroll instantly at the end of making your reservation.

**CONTINUE**

Admirals Club DealFinder Careers Site Map Legal Privacy Policy Customer Service & Contingency Plans Optional Charges Copyright RSS

AAdvantage Earn Miles. Enroll Now. Preferred and Preferred Plus Seats Exclusive Email Offers Fly for The Cure. Donate today!

Figure 2: Example of Test Case

#### 4. Discussions

So Software testing is important as it cause mission failure, impact on operational cost performance and reliability, if it is not done properly. Most of the software programmers expect 100% of quality in the software. So the effective software testing only can make 100% quality softwares. In this research, illustration is made for that how to improve the quality of the software which is used Airlines Management System by efficiently designing the test cases during the test life cycle.

#### Acknowledgements:

Authors are grateful to the Department of Computer Science and Engineering, KSR College of Technology, Tiruchengode, Tamilnadu, India for every support to carry out this work.

#### Corresponding Author:

Nirmal Kumar A

Assistant Professor

Department of Computer Science and Engineering

Christian College of Engineering and Technology

Oddanchatram, Tamilnadu 624619, India

E-mail: [nirmal15483@gmail.com](mailto:nirmal15483@gmail.com)

#### References

1. Poulding, S. and Clark, J.A. "Efficient Software Verification: Statistical Testing Using Automated Search", IEEE Transactions on Software Engineering, Vol. 36, Issue: 6, pp. 763 – 777, 2010.
2. Nidhia, K., Shashank, S. and Ahmed. P. "Gathering Requirements for Hospital Magement System using Intelligent Agents", International Journal of Engineering and Innovative Technology (IJEIT), Vol 1, Issue: 3, pp. 276-279, 2012.
3. Mouchawrab, S., Briand, L.C., Labiche, Y. and Di Penta, M. "Assessing, Comparing, andCombining State Machine-Based Testing and Structural Testing: A Series of Experiments", IEEE Transactions on Software Engineering, Vol. 37, Issue: 2, pp. 161 – 187, 2012.
4. Moonzoo, K., Yunho, K. and Hotae, K. "A Comparative Study of Software Model Checkers as Unit Testing Tools: An Industrial Case Study", IEEE Transactions on Software Engineering, Vol. 37, Issue: 2, pp. 146 – 160, 2011.
5. Mesbah, A., van Deursen, A., and Roest, D. "Invariant-Based Automatic Testing of Modern Web Applications", IEEE Transactions on Software Engineering, Vol. 38, Issue: 1, pp. 35 – 53, 2012.
6. Manjit, K. "Comparative Study of Automated Testing Tools", International Journal of Computer Applications (0975 – 8887), Vol 24– No.1, 2011.
7. John, Ø. and Abdul, S. "Hospital quality management system in a low income Arabic country: an evaluation", International Journal of Health Care Quality Assurance, Vol. 19, Issue: 6, pp. 516 – 532, 2006.
8. Janna, A.F., Kathy. E., Ann. D., Terry. S. and Kathryn. J.H. "Health services innovation: evaluating process changes to improve patient flow", International Journal of Healthcare Technology and Management, Vol.12, No.3/4, pp. 280 – 292, 2011.
9. Hong Mei, Dan Hao, Lingming Zhang, Lu Zhang, Ji Zhou and Rothermel, G. "A Static Approach to Prioritizing JUnit Test Cases", IEEE Transactions on Software Engineering, Vol. 38, Issue: 6, pp. 1258 – 1275, 2012.
10. Hall, T., Beecham, S., Bowes, D., Gray, D. and Counsell, S. "A Systematic Literature Review on Fault Prediction Performance in Software Engineering", IEEE Transactions on Software Engineering, Vol. 38, Issue: 6, pp. 1276 – 1304, 2012.
11. Denver, S., Taryn, A., Shannon, E. and Cheryl C. "Hospitality in hospitals?", International Journal of Contemporary Hospitality Management, Vol. 20, Issue: 6, pp. 664 – 678, 2008.
12. Ali, S., Briand, L.C., Hemmati, H. and Panesar-Walawege, R.K. "A Systematic Review of the Application and Empirical Investigation of Search-Based Test Case Generation", IEEE Transactions on Software Engineering, Vol. 36, pp. 742 – 762, 2010.

5/11/2013