VOL. 10, NO. 3, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

INTERACTION TESTING FOR AN AD-HOC SYSTEM (QUEUE MANAGEMENT SYSTEM)

Ramli R.[1], Mohamed Zabil M. H.[1], Lim K. C.[1],Yusoff Y. and Yusuff M. S.[2]
[1]College of Information Technology, Universiti Tenaga Nasional, Jalan Ikram-Unten, Kajang, Selangor, Malaysia
[2]Department of IT and Multimedia Services, Universiti Tenaga Nasional, Jalan Ikram-Uniten, Kajang, Selangor, Malaysia
E-Mail: ramona@uniten.edu.my

ABSTRACT

The number of student intake to the university has increased significantly for the past few intakes. In normal condition, newly registered student needs to go through various registration related processes before the registration could be completed. The increment in the number of students has affected the whole registration process that could take hours to complete due to bottleneck condition at some counters. To expedite and smooth the registration process, the Queue Management System (QMS) which is used during the registration day has been revised. In order to increase the confidence level of QMS to be used in real environment setting, the system has been tested using an interaction testing technique to detect as many errors as possible due to interaction. Although the testing activity was done in limited time with inadequate requirement and design documentation, quiet a number of errors have been managed to be detected. During the real environment, no system error was recorded and as the result, the whole registration process takes less time compared to the previous years. The results obtained showed interaction testing is possible to be used in detecting faults given time constraints and lack of requirement documentation. This paper reports how interaction testing is designed and conducted as well as discussing the challenges faced by the researches and strategies to overcome it.

Keywords: interaction testing, test data generator, queue management system.

INTRODUCTION

In recent years, the university has seen a significant increase in student intake for each semester. In order to enhance and maintain customer satisfaction, the management has taken an initiative to improve the process flow of new student registration. The students' registration process which involve various departments could last for hours to be completed. The process includes activities such as pre-registration, document verification, accommodation invoice, payment process, issuing Student ID, briefing from the Student Council, taking a survey and ends with students checking into their apartment.

In order to maintain and enhance customer satisfaction, a new registration process flow has been designed and implemented for the upcoming session's intake. The improved process flow is intended to increase the efficiency of the registration process. It is expected that the time taken for a student to complete the registration process will be significantly reduced to around only an hour compared to up to four hours previously. As a critical part of the new process flow, a new Queue Management System (QMS) has been developed.

However, since the system is an ad-hoc system (i.e. rapid development) by the university ITMS department, it has to be tested somehow to increase the confidence level that the system will not fail during the registration day. The QMS is categorized as critical since it will be the first system encountered by the new students. Failure of the system will portray a bad perception to the new students as well as their family and friends towards the university, thus could ruins the university reputation. QMS will filter students, according to a set of conditions and produce a queue number which lead students to next specific counter for next process. Should the QMS fail to produce the correct queue number; student might be led to a wrong counter and this could cause a delay to the registration process. Worse come to worst, if the queue number cannot be produced due to system failure, the respective registrar's officers need to perform manual document checking before directing students to the correct counter. These failures could lead to bottleneck and cause significant delays in the whole registration process.

To reduce the risk of failures, the QMS system needs to be tested adequately to ensure system reliability. The critical part of the QMS is to evaluate a set of conditions before producing queue number. These conditions are constructed from a combination of values of selected table fields from the system's database. These conditions will become the basis for the system in producing queue number for a particular student. Since QMS is an ad-hoc system, time allocated for testing activity was less than a month before the registration day. Given constraints of time and resources, interaction testing is identified as one of suitable testing technique to be performed. This is due to the QMS that involves many combinations of values in order to produce the queue number. Interaction testing will detect faults due to parameters or input interaction (Kamal Zuhairi et al. 2008). By adopting interaction testing, it ensures QMS to cater all possible combinations of conditions. In order to achieve this, an adequate test data need to be prepared. A simple script will be used to automate the interaction testing.

INTERACTION TESTING

Interaction testing detects faults due to interaction between parameters (Kobayashi, Tsuchiya, and Kikuno, 2002; Nie and Leung, 2011; Kuhn et al. 2009). Testing all

possible combination or interaction between the parameters may detect unexpected defects that may resuls from the interaction. It is also proposed that interaction testing is a suitable method to be used within a limited cost and constrained schedule, in order to fulfill the release time and ensure good testing coverage (Huller 2000; Hagar et al. 2014).

The basis of combinatorial testing is the interaction rule between the parameters. The combinatorial testing is also known as Combinatorial Interaction testing (CIT) (Nie and Leung, 2011). CIT is claimed as the effective software testing technique and has been well studied and applied for over 20 years (Nie and Leung, 2011; Cse, 2014). In CIT, not every parameter contributes to every failure (Kuhn et al. 2009). Most failures are triggered by a single parameter value or interactions between a relatively small numbers of parameters.

Pairwise Testing claimed as the most basic combinatorial testing strategy (Kuhn et al. 2009).In this strategy, a set of test cases is generated to covers all combinations of the values for each pair of parameters. Pairwise testing normally is referred as all pairs testing, pair testing, pair-wise testing and 2-way testing (Bach and Schroeder 2004). Number of test cases can be reduced by using this strategy since exhaustive testing might not be possible to conduct due to time and resource constraints.

Compared to another popular method of CIT, Orthogonal Array (OA) strategy (Banerji 2012), pairwise testing is considered better (Hunter 2013). Fewer test cases are required compared to OA. Thus, this can reduce the testing effort and cost meanwhile ensuring the efficiency of testing.

On top of that, results from empirical studies comparing combinatorial testing with manual test case method shows that more additional defects were found using combinatorial testing. Large number of defects found in limited amount of time. Number of test case required for the testing coverage are also reduced. (Kuhn et al. 2009; Khalid et al.; Hexawise 2014).

Several pairwise tools available commercially for developing the test case generation (Czerwonka, 2014; Hexawise Team, 2012; Udai 2014). Each of the tools differs from its characteristics and advantages. From the researchers' knowledge, no evaluation has been done to know which tools works suitably for which settings.

For this study, based on survey done by the researchers, Hexawise (Hexawise Team, 2012) was choosen as the test case generation tools. The choosen was based on empirical study conducted by IEEE (Hexawise, 2014; Kuhn et al. 2009) that using Hexawise test case can be generated faster compared to manual. The generated test cases are more detailed and efficient. Through Hexawise, minimum number of test cases are generated to achieve maximum test coverage. In addition, the tool is claimed being used widely by many Fortune 500 companies (Hexawise Team, 2012).

The interaction testing works by matching all possible combinations of parameter values involved. For an illustration, should we have a system with four input parameters, A, B, C and D. Each parameter has 2 possible values indicated as 0 and 1. The parameters and its values are described as in Table-1 below.

Table-1. A system with 4 input parameters and its values.

Parameters	A	B	C	D
Values	a0	b0	c0	d0
	a1	b1	c1	d1

If the test objective is to cover 100% pairwise interaction, for each parameter pair, all interaction sets for the pair need to be covered at least once as in Table-2 below.

Table-2. Pairwise Interaction sets for 4 parameters.

Parameter interaction	Interaction set to be covered
AB	a0b0, a0b1, a1b0, a1b1
AC	a0c0, a0c1, a1c0, a1c1
AD	a0d0, a0d1, a1d0, a1d1
BC	b0c0, b0c1, b1c0, b1c1
BD	b0d0, b0d1, b1d0, b1d1
CD	c0d0, c0d1, c1d0, c1d1

Finally the complete test cases for pairwise interaction test need to be generated. For each parameter value, it needs to be tested with all other parameter values at least once. Table-3 below illustrates the example of test cases (i.e. un-optimized) for pairwise interaction for the illustrated system.

Table-3. A system with 4 input parameters and its values.

Test cases	A	B	C	D
TC1	a0	b0	c0	d0
TC2	a0	b0	c0	d1
TC3	a0	b0	c1	d0
TC4	a0	b0	c1	d1
TC5	a0	b1	c0	d0
TC6	a0	b1	c0	d1
TC7	a0	b1	c1	d0
TC8	a0	b1	c1	d1
TC9	a1	b0	c0	d0
TC10	a1	b0	c0	d1
TC11	a1	b0	c1	d0
TC12	a1	b0	c1	d1
TC13	a1	b1	c0	d0
TC14	a1	b1	c0	d1
TC15	a1	b1	c1	d0
TC16	a1	b1	c1	d1

As we can see, for a system with 4 parameters and each parameters has 2 possible values, in total there are 16 test cases need to be executed for pairwise interaction testing.

THE STUDENT REGISTRATION PROCESS FLOW

Prior to the implementation of the new QMS, all new students have to go through a general counters for

registration. The registration process for students with incomplete documents is significantly longer compared to those with complete documents (i.e. Completed payment, and all online procedures). In this scenario, should there be many students with incomplete documents; other students who have completed their document will also be affected in terms of long waiting queue before they will be served. As the number of new students increase in every intake, there is a need to improve the registration process to avoid long waiting time and create inconvenience among students and parents.

In order to improve the process, several categories of students are studied to determine their conditions that will lead them to dedicate counters, based on their academic
program, type of students (i.e. Local or international) and their registration status (i.e. Complete or incomplete).

Multi-channel and multi-phases system concept (Stevenson and Hojati 2007) is used for this purpose. In this concept, there is one waiting line, but several counters provide specialize service. For a particular customer, he or she receives services from several stations before ending the service.

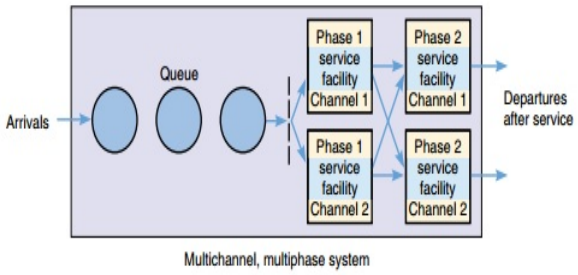

Figure-1. Multi-channel and multi-phases system concept (Stevenson and Hojati, 2007).

Back to the QMS, five counters are identified to handle the registration process according to the scenario.

Table-4. Counters for registration process.

Counter name	Scenario
A	Mainstream Express Students
B	Mainstream Normal Students
C	Walk in Students
D	International Students
E	Special Program Students

The registration process begins with checking the student's record based on their National Registration Identity Card (NRIC) or passport number. This step determines the next counter that will serve the student. Students will be issued a relevant queue number based on their application record. The queue number leads the flow of the registration process.

For the newly designed registration process flow, the estimated maximum time for the whole registration process is less than one hour. Thus, to ensure the system work accordingly on the actual day, thorough testing needs to be conducted. The smoothness of the process on the registration day will avoid customer's frustration and enhanced the quality of administrative work itself.

THE LOGICAL DESIGN OF QUEUE MANAGEMENT SYSTEM (QMS)

QMS system is designed to handle five categories of students during UNITEN's registration day. The objective of the system is to issue a queue number that leads to a dedicated counter based on certain condition of the system. The conditions are based on some combinations of data fields imported from UNITEN's online application system.

The database of the online application system consists of two main tables (i.e. ApplicantOnline (AO), ApplicantPayment (AP)) and an auxiliary table (i.e. ApplicantPaymentType (APT)). Table APT stores the payment type ID which is used in table AP. Table AO contains applicants' data, such as applicant's personal information (e.g. Name, NRIC, address, contact information
and etc.), academic details (e.g. Education background and exam results), programs applied by the applicant, program offered by the university, and status of the acceptance of the offer.

Table-2 AP stores applicant's payment information and its status. In table AP, payments made by an applicant related to the application and registration process are recorded. Since table APT is an auxiliary table, only table AO and AP are considered in generating queue number. Both table AO and AP have one-to many relationship as illustrated in Figure-2. The applicant's record is unique in table AO, on the other hand, there will be many payment transaction records in table AP for any one applicant.

From both tables, several attributes are identified as the QMS parameter to determine the scenario for each counter. A dedicated queue number will be issued based on the identified scenario.

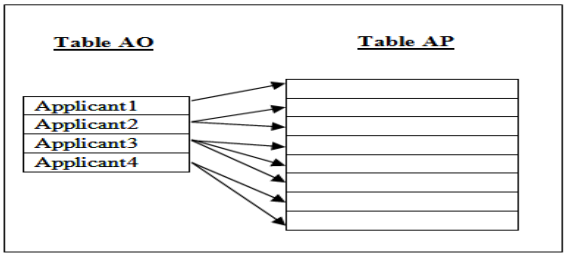

Figure-2. The one-to-many relationship between table AO and table AP.

For counter A, B and C, the same parameters are identified. However, values of each parameter will differentiate the counter that leads the student's registration process. International students will be handled

by counter D due to special conditions for them. The only attribute to differentiate this are the nationality ID. Counter E is dedicated to students for special academic programs since no payment will be involved during the registration process.

Table-5. QMS parameters that are considered in generating queue number.

Counter	Table AO	Table AP
A (Express)	Finalized completed offer accept Student ID	Finalized Payment type ID
B (Mainstream)		
C (Walk in)		
D (International)	Nationality ID	Not applicable
E (Special Program)	Is foundation	Not applicable

With regards to QMS, the system will receive either Applicant's Online ID (AOID) or NRIC No as the input Figure-3. Generation of the queue number will depend on the input. The queue number is in 4-digit form, where the first digit will indicates the counter Figure-4.

The ability to generate queue numbers is the critical part of the QMS. In order to ensure the system will not fail during the registration day, extensive testing before the registration day is needed for this purpose. Indirectly, the prior testing will facilitate the registration process and improve the customer services. Further details on the test design and test data generation process of the QMS discussed in the next section.

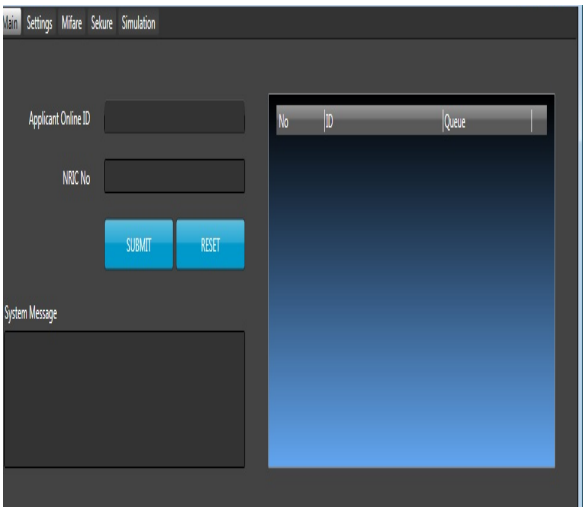

Figure-3. QMS receives input.

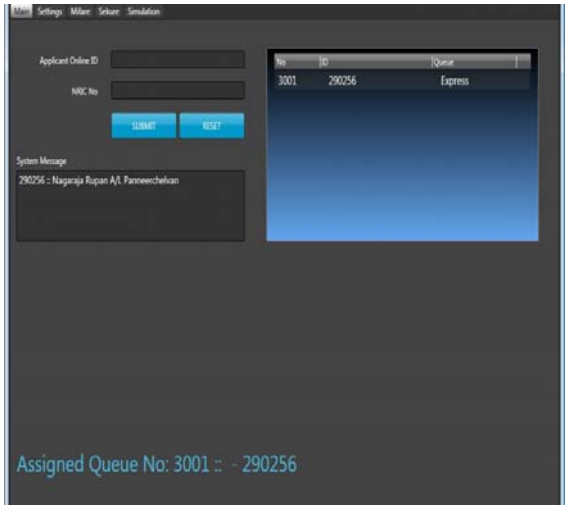

Figure-4. The generation of queue number.

TEST DATA GENERATION FOR QMS

In order to generate test data for QMS, we must first identify the inputs involved in generating the correct queue number. Based on our analysis, we divided the system's inputs into two categories. We refer the two categories as External and Internal inputs. The External inputs are inputs that are given by users via the system's interface. There are two inputs/parameters involved, namely, AOID and NRIC number. A user has to key in either one or both through the system's interface. Based on these inputs, the system will look into the database for a match and subsequently check which category does this student's belongs to. Thus, we group the inputs obtained from the system's database as Internal inputs.

Test data for external parameters

There are two External inputs for QMS: AOID and NRIC. The AOID is generated when applicants submit their application online, in the form of 6-digit number. The NRIC is the applicant NRIC or passport number. For both inputs, the expected results are either Found or Not Found.

Table-6. External inputs.

Parameter	Description	Expected results
AOID	The applicant online ID is generated when applicants submit their application online, in the form of 6-digit number	Found Not found
NRIC	The applicant NRIC or passport number	Found Not found

In searching for an applicant, user can key in either AOID or NRIC. The system then will check whether there exist a record for the particular AOID or NRIC. Four test cases were derived, which covers valid and invalid situation.

Table-7. Test Case for External parameters.

TCID	Test case
TC01	Key in a valid AOID
TC02	Key in an invalid AOID
TC03	Key in a valid NRIC
TC04	Key in an invalid NRIC

Test data for internal parameters

As discussed in previous section, we have identified seven parameters from table AO and two parameters from the table AP to be considered for generating test data for interaction testing. The selection of these parameters is based on the information given by the developer, the study of business rules as well as performing a simple review of an important part of the system's source code.

The parameters considered for test data from table AO are Finalized, Completed, Offer, Accept, StudentID, Nationality and isFoundation. Each of the parameters have their own values.

Table-8. Internal parameters from table AO.

Parameter	Values to be tested
Finalized	0 or 1
Completed	0 or 1
Offer	0 or 1
Accept	0 or 1
StudentID	Null or not null
Nationality	Local (100) or international
isFoundation	0 or 1

The parameters considered for test data from table AP are Finalized and PaymentTypeID with their respective value, either 0 or 1.

Table-9. Internal parameters from table AP

Parameter	Values to be tested
Finalized	0 or 1
PaymentTypeID	0 or 1

The business rule on how applicants are being sorted according to values in both tables are shown in Table-7.

Table-10. Summary of conditions to produce a queue number.

Counter	Table AO value	Table AP value
Express	Finalized=1 Completed=1 Offer=1 Accept =1 StudentID not null	Finalized =1 PaymentTypeID = 4
Mainstream	finalized=1 completed = 1 offer=1	Don't care
Walk in	isCFGS=false no record OR finalized =0 completed=0 offer=0	Don't care
International	Nationality=100	Don't care
Special program	isFoundation=1	Don't care

Since there are five service counters and each service counter has its specific condition, the test data will be generated based on one counter at a time. Since the QMS is a critical system, we decided to generate the interaction test data at full strength. This is still practiced since the number of parameters involved for each service counter is low. On top of that, the possible values for each parameter are minimum (i.e. Maximum value is two). Besides that, since all inputs are internally obtained from the database the test execution will be done automatically using scripts.

Based on these factors, full interaction strength is still practical for QMS. Furthermore, we can assure the effectiveness of the testing as full interaction is used, thus increased the confidence towards the system.

Since each service counter has different conditions, test data was generated separately. Once all the test cases have been generated, the test case data are imported into the QMS database. At this point the QMS database consists of all test data (i.e. Students' record) with all possible scenarios to be tested. From Table-5 and Table-6, the parameter configuration for interaction testing was produced. Subsequently, the interaction testing, test data is generated using online tools (Hexawise Team 2012)The parameter configurations as well as the number of test cases are shown in Table-8.

Table-11. Configuration for interaction testing.

Counter	Configuration
Express	7 parameters, each parameter has 2 values Total test cases: 32
Mainstream	3 parameters, each have 2 values Total test cases: 32
International	1 parameter with 2 values Total test cases: 7
Special program	1 parameter with 2 values Total test cases: 32
Walk in	Scenario 1 (online transaction not finalized) 3 parameters, each have 2 values Total test cases: 16
	Scenario 2 (no record and not Foundation program) 1 parameter with 2 values Total test cases: 16

Test execution

For the External parameters, the test is done manually. On the other hand, for the Internal parameters, we automate the test execution using scripts. Each test case represents a student's record. Therefore, random NRIC (for local students) and passport number (for international students) are added to each test case. This will allow the script to simulate the QMS database lookup process. Figure-5 describes the logical flow of the script. The script starts by selecting all records from the respective tables (AO and AP). For each record, the script will invoke the QMS functions to evaluate the conditions. Based on the provided data, QMS produces the appropriate queue number for the students. This queue number is stored for comparison.

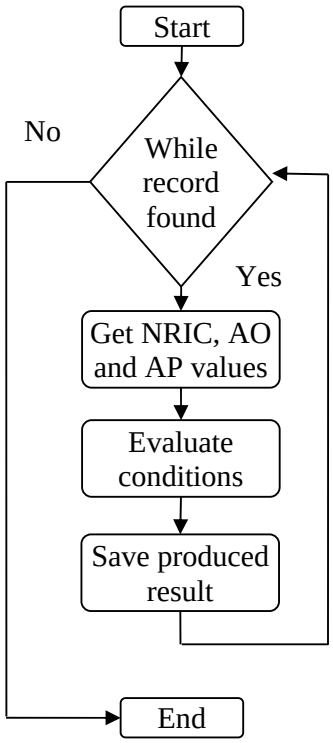

Figure-5. Logical flow of the script for test automation.

After executing the test script, two interaction errors were found. The logical errors are related to the condition for producing queue numbers in the QMS. After the logical errors have been fixed, we run the test again for several more times and the results from QMS are similar to the expected results. Although no critical fault was found with regards to the QMS, the testing process has been very useful in increasing the confidence of the development team in deploying the system.

Manual observation was conducted during the registration day to ensure no defects will occur in the real setting environment. From the observation, no defects were reported found. The registration process runs smoothly and without hiccups. As the result, the average time of the registration process need to be completed for a student takes only one hour compared to the timing for previous intakes. Indirectly, the results has validate the confidence level of QMS gained during the testing activity.

RESULTS AND DISCUSSIONS

Although there are many aspects of the QMS that could be tested, but due to time constraint (time to deploy) the researches only managed to perform interaction testing tests, to ensure faults due to a combination of conditions, that is the main part of the system, can be detected. On top of that, being an ad-hoc system, QMS is lack of proper documentation (e.g. Requirement and design documents). This limits the type of testing that can be performed to test

the system due to there are no point of reference whether the system perform as it required or otherwise. In our case, the oracle (i.e. Expected output) is based on the business rules of the registration process.

The strategies used by the researches to deal with the challenges are discussed as follows:

Challenge 1: Inadequate test basis

One of the challenges faced by the researches in this study is to understand the system requirement with inadequate test basis. The walkthrough technique(ISQTB 2011) was applied to understands the business rules of the registration process. Several session were conducted between the development team and researchers, which are led by the author. During the session, researches gained better understanding on the business rules. The session had helped the researches in generating the test cases within limited time. The cooperation from the developer team also had contributed to the successfulness of this study.

Challenge 2: Limited time to design test data

Designing test cases using the pairwise testing strategy can be tedious and takes a lot of time and effort. Thus, selection of right testing tools to design a test data, plays an important roles. Hexawise(Hexawise Team 2012) was chosen in this study based on its features that can generate test data within limited time and at the same time ensure the good testing coverage.

CONCLUSIONS

This paper presented the researches' work in generating test cases for interaction testing. The test data is to be used for testing the QMS, which is an ad-hoc system. The test cases were designed based on the business rules provided by the developer team.

The pairwise testing was chosen as the test design strategy. The chosen was made based on its ability to generate test cases that cover all combinations of the values for each pair of parameter. Considering that the researchers were given limited time to complete the activity due to the short release time, several pairwise testing tools were considered to be used as the test data generator.

Hexawise (Hexawise Team 2012), a web-based, free and commercial tool was selected as the test data generator tool. Test data covers all pair of interaction and ensure good testing coverage. The tools also helped the researches to generate test data within limited time.

The design of the test cases were also discussed in detail in this paper. A total of 153 test cases were generated. The execution of the test scripts showed that only two interaction defects were occurring. The logical errors only occurred related to the condition for producing the queue numbers in the QMS. After the logical errors have been fixed, the test script was executed again for several times. From the execution, results from QMS are similar to the expected results. No critical fault was found with regards to the QMS, thus this had increased the confidence of the development team in deploying the system.

From the manual observation made by the development team during the real setting, no defects were reported found. The registration process runs smoothly and without hiccups. Indirectly, the confidence level of the QMS has been confirmed.

Strategies used by the researchers to deal with the challenges during the testing activity were also discussed. On top of the challenges, the research team manages to complete the testing activity and increased the confidence level of the development team to release the system.

ACKNOWLEDGEMENTS

This study was a collaboration of short-term project between College of Information Technology and Department of IT and Multimedia Services (ITMS), which both teams are from Universiti Tenaga Nasional.

REFERENCES

Bach James. and PJ Schroeder. 2004. "Pairwise Testing: A Best Practice That Isn't." In Proceedings of 22nd Pacific Northwest Software. pp. 175–92.

Banerji Shubhra. 2012. "Orthogonal Array Approach for Test Case Optimization." International Journal of Advanced
Research in Computer and Communication Engineering 1 (9): 613–21.

Cse UNL. 2014. "Combinatorial Interaction Testing Portal." http://cse.unl.edu/~citportal/.

Czerwonka Jacek. 2014. "Pairwise Testing - Available Tools." http://www.pairwise.org/tools.asp.

Hagar Jon., Rick Kuhn., Raghu Kacker. and Tom Wissink. 2014. "Introducing Combinatorial Testing in a Large Organization: Pilot Project Experience Report." 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops, March. Ieee, 153–153. doi:10.1109/ICSTW.2014.70.

Hexawise. 2014. "Does Pairwise Testing Really Work? Evidence, Data, and Case Studies." http://training.hexawise.com/m/7455/l/74099-9-does-pairwise-testing-really-work-evidence-data-and-case-studies.

Hexawise Team. 2012. "Hexawise–Pairwise Testing Made Easy."

Huller Jerry. 2000. "Reducing Time to Market with Combinatorial Design Method Testing." In Proceedings of the 2000 International Council ….

Hunter John Hunter., Justin. 2013. "Which Is Better Orthogonal Array or Pairwise Software Testing?"

https://hexawise.com/posts/which-is-better-orthogonal array-or-pairwise-software-testing.

ISQTB. 2011. Certified Tester Foundation Level Syllabus.

Kamal Zuhairi M. I., Younis., Syafrul Afzal Che Abdullah. and Zainal Hisham Che Soh. 2008. Software Testing. First Edition. Kuala Lumpur: Open Universiti Malaysia.

Kobayashi Noritaka., Tatsuhiro Tsuchiya. and Tohru Kikuno. 2002. "A New Method for Constructing Pair-Wise Covering Designs for Software Testing." Information Processing Letters 81 (2): 85–91. doi:10.1016/S0020-0190(01)00195-8.

Kuhn Rick., Raghu Kacker., Yu Lei. and Justin Hunter. 2009. "Combinatorial Software Testing." IEEE Computer Society, August. doi:10.1109/MC.2009.253.

Nie Changhai. and Hareton Leung. 2011. "A Survey of Combinatorial Testing." ACM Computing Surveys 43 (2): 1–29. doi:10.1145/1883612.1883618.

Stevenson WJ. and M Hojati. 2007. Operations Management. Pearson Education Limited.

Udai Shekhar. 2014. "A Literature Survey on Combinatorial Testing." International Journal of Advanced Research in Computer Science and Software Engineering. 4 (4): 932–36.