Adaptive Testing techniques to reduce Cycle Time - Moving from Theory to Practice

Jitender B Danwani, Ajay K Chhokra
UnitedHealth Group Information Services Pvt Ltd.
Gurgaon, Haryana, India
jbdanwani@yahoo.com, ajay_k_chhokra@uhc.com,

Abstract:
The buzz word among CIOs of Information Systems organization is reducing Cycle time by using latest adaptive testing techniques. To obtain the greatest return on their investment, IS organizations has started putting greater emphasis on the systems testing process and implementing derived best practices based on theoretical techniques like pair-wise testing, cycle based testing and Risk based testing. These theoretical techniques looks great in research paper but gives negative results if one tries to implement without tailoring it to business needs.

Through this paper, we intend to present the need and importance of implementing Adaptive Testing Techniques and share our experience of implementing best practices of Pair-wise testing, cycle based testing and risk based testing to achieve goal of reducing cycle time but marinating test effectiveness. The test teams implementing these adaptive testing techniques need to understand the theory, merits and limitations and spend good amount of time in tailoring it to project needs without losing focus on the goal of reducing cycle time and quality effectiveness of testing. The objective here is to achieve complete test coverage in minimum cycle time using minimum number of resources. This requires smart planning and strategizing the efforts. Various testing techniques are being applied to manage this increasing pressure. These techniques attempt to reduce costs by selecting and running only a subset of the test cases in a program's existing test suite. However, no single technique today is able to satisfy all the business needs. The testers across the globe have experimented various testing models in a controlled environment (Pilot Project) that tend to guarantee some degree of quality and customer satisfaction but when projects try to scale up to real environment, the results does not come encouraging especially on Testing effectiveness and ROI. . We in UHG IS, Gurgaon have applied these techniques in our Testing Projects “Mix–n-Match” taking care of their strengths and improving upon their limitations thus giving maximum ROI on cycle time and test effectiveness. In this paper we will be covering our methodologies, case study and results analysis.

There are two aspects of applying these adaptive techniques – one in terms of defining them and the other more challenging is implementing across the projects with Metrics built around to capture the score card. The metrics is very critical to the success as it helps towards refining it till the target goals of cycle time improvements are achieved.

Keywords:
Risk Based Testing, Pairwise Testing, Reduce Cycle Time

1 Background

Today, there are lots of reduction techniques available in market which has been successfully applied across projects, domains and IT companies. These techniques are been researched or developed to help projects reduce the time spent in performing test execution for identified system functionalities. These techniques are commonly referred to as ***'Reduction techniques'*** because they help a tester to reduce the overall test efforts by reducing the number of test cases required to test business functionality. These reduction techniques not only reduce the case count but also focus on other quality parameters such as – high test coverage, reduce the business risk, reduce the time-to-market etc. Some of the familiar techniques prevailing in market today are – Risk *Based Testing, Pairwise testing and Cycle based testing*. Each technique is unique in itself and bears some advantages and disadvantages. However, for a tester, the bigger challenge is to answer some of these questions - Can a technique be applied and used for any kind of project? Do these techniques have any limitations? Is there any major business impact when applying any of these techniques?

2 Reduction Techniques

Let us first discuss on various reduction techniques that are commonly available in market, their merits, shortcomings and suitability. We will then discuss on 'Adaptive techniques' and their suitability.

2.1 RISK BASED TESTING

Before, we look into what is the technique about, let's first define – what is Risk? A risk may be defined as threat to the successful achievement of a project's goals. It is a factor that could result in future negative consequences. A tester's job is to find out high risk areas and identify any associated bug.

2.1.1 *What is Risk Based Testing?*

Risk Based technique is one of the most popular testing techniques and is widely used in the IT companies. The technique focuses on analyzing and understanding the business risk. It involves identification, and measurement of the business critical functions in the application domain. Through this technique, a quality analyst tends to analyze application or interfacing area which have maximum impact on the business or probability of failure.

A tester, to some extent, use the risk based technique (knowingly or unknowingly) to identify and analyze the risk prone areas. However, the approach taken by him is more like an ad-hoc exploration of business areas rather than rationally defined. It is more intuitive than calculative. A risk based technique is significant if it can determine and ensure the quality of product quantitatively using metrics.

2.1.2 *Need of Risk-Based Testing*

The testing is generally considered as tester's job. Most testing is based on tester's judgment and his ability to find defects rather than any specific technique. There are certain things you must do. Those things vary depending on the kind of project you're on, your industry niche, and so on. Risk is a problem that might happen. The more likely the problem is to happen, and the more impact it will have when it happens. As the impact increases or probability of reoccurrence increases, the overall

risk associated with product increases. Thus, testing is someway governed by risk associated with a product. So, in short, tester's job is to minimize the risk associated with any product or its feature. This makes it imperative to apply risk based technique while doing testing.

2.1.3 *How to do Risk Based Testing?*

One of the common questions that tester have while applying this techniques is how to determine the risk and its probability. The process is simple if you can answer the following questions:

• Which functions and attributes are critical (for the success of the product)?
• What would be the impact of the code change or functionality on stakeholders? (For customers, users, people outside)
• What is the frequency that a particular code will be hit or function will get accessed?
• What will be the impact if function is removed or not working as desired?

Identifying the business critical functionality is not a difficult job. The best way to get answer to the above questions for identifying the critical business areas is through discussion with the key project stakeholder and the business sponsor, person who is driving the project. However, over a period of time through the domain knowledge and experience with testing, a tester may be able to identify the right area with reasonable amount of efforts. Some ground areas that can aid in determining the risk to project are listed below.
–Complexity of application
–Extent of change by means of defect fix or enhancement
–New technology and tools
–Reduced Timelines
–Number of defects found
–History of prior use

2.1.4 *Merits & Shortcomings*

Though Risk Based technique helps to identify key business functions to be tested, the technique cannot be applied in all kinds of projects. There are certain limitations of this technique. Some of the key advantages and limitations of risk based techniques are listed below:

Merits of Risk Based Testing

- Identifies and focus on the key business areas
- Helps to quantify the business functions based on cost and probability
- Eliminates all the major risk from in the product delivery, if planned accurately
- Quick to apply if the tester and business have good exposure to business domain

Shortcomings of Risk Based Testing

- Identifying the business risks precisely is difficult. It requires clear understanding of applications and business and detailed knowledge of requirements
- Technique focuses on the key risks and often ignores the low impact areas.
- The technique is not fool proof; it is more or less intuitive in nature. The impacting functionalities are identified based on tester's judgment. It doesn't completely eliminate the possibility of defects...

- Normally, the technique doesn't provide 100% test coverage. It doesn't guarantee if all the possible interactions between impacting and non-impacting parameters are covered.

2.1.5 *Measuring Risk*

A common question that many testers have asked me is – what are the measurement criteria for risk based testing? How do we measure risk and how we can use the statistics significantly in the projects? Generally; in risk based testing, one of the important measurement criteria is - degree of impact. We can look into the following parameters to determine the degree of impact:

1. *Cost of Impact*
   – Financial damage, customer dissatisfaction
   – Impact on other functions or systems
   – Detection and repair time

2. *Probability of risk or failure*
   –Complexity of system,
   –Knowledge of the application/ interface

If you want higher confidence that you are testing the right things at the right time, risk based testing can help. It focuses and justifies test effort in terms of the mission of testing itself. Use it when other methods of organizing your effort demand more time or resources than you can afford.
If,

R (f) - Calculated risk of function
P (f) - Probability of a fault in function
C (f) - Cost related to a fault in function

Risk = Impact * Probability or Calc risk

2.1.6 *Suitability*

Primarily, the Risk Based Testing is an intuitive technique that if governed by tester's knowledge of business, domain, technical expertise and analytical skills. It cannot be applied and used for all kinds of projects and applications. We will now discuss on the various factors that need to be considered before making a decision whether or not to use the risk based testing.

1. *Complex applications*

When the application or system under test are complex, which means there are too many applicable business logics or conditions etc, the amount of time it takes to determine the risk and cost associated is too much. At times, it is not possible to accurately determine the complete impact and associated cost. In such cases, applying risk based testing is costlier affair as the investment is much more than the return. Risk based testing is well suited for small and medium sized projects where all the conditions and their impact can be easily determined.

2. *Project Delivery Cycle*

Every project has a defined delivery schedule (also referred as "Project Schedule"). The delivery schedules and milestone dates are defined during test planning phase at start of project. Many a times, the delivery cycle are shrunk (cut-short) when the projects is under estimated or there are delay from previous deliveries, the unforeseen business delays or inter-dependency delays. This

results in a direct/ indirect impact on the project life cycle. Testing being the last stage of project lifecycle has to bear this impact. Risk Based Testing is often a preferred technique in such scenarios when the delivery cycles are small or cut-short. The technique helps to identify and focus on all the business critical functions that must be tested. It helps to reduce the burden from testers in determining and prioritizing the testing efforts.

2.2 PAIRWISE TESTING

2.2.1 *What is Pairwise Testing?*

Pairwise testing is a market proven technique that is based on Orthogonal Arrays Concept. This technique as name suggests – "pair up" - involves pairing up of various parameters (aka factors) that affect a business functionality to form unique combinations that define a test case.

Pairwise Concept

- Any business requirement can be segregated into functionalities and sub-functionalities.
- Each functionality can be a function of various parameters f(x) = y(a, b, c)
- The variable/s parameters may take multiple values called as Levels
- The various combinations of factors and levels help derive exhaustive set of test cases.
- Pairwise technique helps identify unique combination of these factors / levels to arrive at minimal set of test cases that can still ensure maximum coverage.

2.2.2 *Need for Pairwise?*

As a tester, you might sometime find yourself in situations when a project is assigned to test all possible functionalities of an application without losing the focus on quality. You have been asked by your manager to ensure that product delivered is defect free. Add to it, is the challenge of testing various branches, high risk modules, various combinations of interacting factors within the defined timelines. The delivery deadlines are generally robust and defined much ahead of project allocation. Since the number of possible inputs is typically very large, testers need to select a sample, commonly called a suite of test cases, based on effectiveness and adequacy. You take all precautionary steps to ensure that you don't miss any functionality, or test conditions or interactions among two or more variables. Much functional testing is done in an intuitive and less formal manner, so there is the possibility that important interactions among the inputs will be missed.

E.g. If you are asked to test a screen with 50 fields, how many test cases you need for exhaustive testing? Assuming only 2 values per field (like "Status": Inactive or Active) per field, one has 2^{50} test cases. Even with automated testing, at a rate of one million cases per second, this will require $2x10^{15}$ years to test all possible cases.

In such scenarios, we look forward to apply some techniques that can help us offload our work of analyzing the application functions and identifying precisely what are the interacting variables or parameters. Pairwise testing is one such technique which can answer this question.

2.2.3 *How to apply pairwise testing?*

As Pairwise testing is based on Orthogonal Array concept, we need to first understand the concept of Orthogonal Array which is beyond the scope of this paper. However, I will just mention the three ways that are commonly used by testers.

a) *Manual Method*: In this way, testers apply the orthogonal concept manually to derive the various matrixes at different interaction level. This is a tedious and time consuming task. It require skilled person who have deep statistical knowledge to learn and implement the concept.

b) *Using Freeware Tools*. – There are number of freeware tools available in market (such as AllPairs etc) which can easily help arrive at Pairwise matrix. The tools are not flexible and cannot be customized. They can provide results only at single interaction level and without any parameter constraints being set.

c) *Using Vendor Tools* - These tools are vendor supplied and have cost associated with them. However, these are highly customizable and can help arrive at matrix at single or multiple interaction levels. The response time for the tools is relatively low compared to other freeware tools and these can also be used to set the business constraints – to include or exclude specific combinations of parameters.

In simple steps, as a tester if you need to understand how to go-about applying the Pairwise testing, you can follow the steps below:

- ✓ Segregate all the business requirements to High Level Functions for the Interface under consideration
- ✓ Identify all possible factors for the identified functions and short list ones to be applied for Pairwise
- ✓ Identify Levels (Values) for each factor
- ✓ For each transaction, map factors and relevant levels
- ✓ Identify the business rules, dependencies and exceptions
- ✓ Input the factors, levels, dependencies to Pairwise testing tool
- ✓ The tool will help arrive at the reduced set of cases

2.2.4 *Measuring Pairwise*

Pairwise testing help arrive at reduced number of test cases. The measurement of results from pairwise testing is not an easy job when it comes to manual analysis. However, using the Pairwise tool, we can arrive at the test coverage and the reduction efficiency at varied interaction levels.

2.2.5 *Merits and Shortcomings*

The Pairwise testing is statistical technique that helps in test case reduction. This technique involves generating all possible combinations of interacting parameters using minimal set of cases. The technique, although appears impressive, is not completely free from limitations. Following are some of the merits and demerits of using Pairwise technique:

Merits of Pairwise Testing

- Help reduce the test cases drastically especially at 2 and 3-level interactions
- Pairwise technique proves helpful for complex systems that involve large number of parameters and their combination
- Pairwise testing can be applied for regression cases where coverage is to be maintained high with minimal cases count.
- The defect detection rate versus the applied effort is high.

Shortcoming of Pairwise Testing

- Some of the defects may go undetected when pairwise testing is applied at 2 or 3 level interaction. There is a possibility of defects at higher order interaction (say 4, 5 level interaction) :
- The number of interaction levels increases, the number of cases increases very significantly.
- Pairwise testing fails when the parameters and levels have not been appropriately detected or recorded...
- Whether to apply single level or multi-level interaction or what order of interaction depends on tester's ability to analyze the requirements and judge possibility of defects. If the selection criterion is not defined correctly, pairwise testing fails.

2.2.6 *Suitability*

The Pairwise testing is statistical technique that is well suited when a system or application require too much of computation to arrive at varied combinations of parameters. Testing exhaustive combinations is not possible always and hence there is a need to apply reduction techniques. This technique is suitable in the following situation.

1. Complex applications

When an application or system under test is complex, there are generally, too many interacting parameters or levels involved within each parameter (generally > 5). In such cases, the exhaustive test case count may turn out to be very high and it may take years to do complete testing. Even when applying risk based technique, determining the impacting parameters and their combinations, risk and associated costs would be time consuming affair. Pairwise testing can be used to reduce the exhaustive test case count by over 90% while still ensuring that all combinations are covered.

2. Exhaustive Coverage (Regressions)

In some project situations, most of the functionalities are business impacting and they are required to be tested in entirety within the defined project timelines. The focus is more on coverage than on specific application nodes. Pairwise testing is best suited in such scenarios as it can effectively generate a reduced set that can ensure high coverage with minimal execution. The Pairwise testing can be successfully applied while performing regression type testing which spans over various release cycles and require high coverage.

3. Short span Cycles:

Pairwise testing is also the most favored technique when it comes to Sanity and Smoke testing. In such projects, a reduced set of test cases is derived at 2-level interaction level. The reduced test cases are used for doing the sanity testing to ensure that basic project functionalities are working as expected

2.3 CYCLE BASED TESTING

Another common technique prevalent in market is the cycle based testing.

2.3.1 *What is Cycle Based Testing?*

A cycle based testing is a technique that involves planning and executing the test cases in cycles. A cycle may simply be defined as test iteration. Every iteration has a defined timeline and the test cases are distributed over iterations based on business impact, priority or test coverage.

2.3.2 *Need of Cycle based technique?*

In last one decade, the orientation toward the testing has changed a lot. Earlier, developers who used to design and code the functionality were also responsible for testing the same. As the applications complexity increased, there was a need for separate test team who can ensure that there are no loopholes in the code. The tester used to test the application in entirety covering all the functions, business logics and interfaces. This was a time consuming affair as the testing proved to be never ending task and spanned over months to years (in some cases). There was a need to cut-short the testing time without compromising on the quality. One of the methods applied was to split different functionalities in multiple cycles. The functions to be tested were selected in each cycle based on business impact. This technique was first used in telecom product-based companies, which had quarterly releases.

Today, the need of cycle based testing is understood by all big IT organizations. The technique helps project manager ensure that the product that span over release gets adequately tested in cycles to meet the quality compliance. Also, the time –to-market the product is reduced drastically as product features can be successfully released cycle-over-cycle.

The cycle based testing is not so commonly known technique like other reduction techniques. However, it bears its own advantages for which it is recommended in specific project situations. It is most suitably used in conjunction with other techniques.

2.3.3 *How to apply Cycle based Testing?*

A cycle based testing technique does not require any special skills. This technique is generally applied when there is enough timeframe for testing. E.g. in a quarterly release if testing spans for a month, then the cases can be distributed over 2 cycles or in some cases 3 cycles. In most projects, the three cycles are assumed to be sufficient to ensure that product/ features are adequately tested. The three basic test cycles are:

a) *Sanity Cycle* - This cycle spans over 2-5 days period and covers the basic test functionalities. The objective of this cycle is to ensure that all the components are intact and working as desired before carrying out full-fledged testing.

b) *Priority Cycle* – In this cycle, all the high priority cases are executed. The priorities are defined based on business impact and Cost of Quality (COQ) that goes in resolving the defect. The defects found in these cycles are triaged and fixed on priority and generally retested in next cycle.

c) *Retest Cycle* - In this cycle, all the low priority cases are being executed. Any defect that could not be closed in Cycle-1 is taken on priority for retest in this cycle. The cycle also ensures that the defect fix on business critical functionality doesn’t impact other non-critical business.

2.3.4 *Merits and De-merits*

Merits of Cycle Based Testing

- Helps to keep a focus by distributing the high-risk functions from low-risk. This is specifically helpful in regressions when cycles are repeated in every release.
- Cycle based testing helps to define logical milestone for testing. The defect fixes or patches applied are tested in subsequent cycles.
- This type of testing is helpful when applying Incremental Integration techniques.

Shortcoming of Cycle Based Testing

- The technique is discouraged when individual test cycles are too long.
- This technique may involve overheads if repeated data setup is required in every subsequent cycle.
- Too many cycles in single release delays the process.
- Tracking the execution, defects and cross-impact is sometimes difficult when using this technique.

2.3.5 *Suitability*

The Cycle based techniques is well suited in the following situations:

4. Incremental Integration Testing

The Integration testing can be conducted using various techniques. One of the techniques is Incremental integration which involves testing the function interactions incrementally. The cycle based testing is helpful to define logical breaks in the integration testing. In case of any bugs found in one cycle, those can be fixed and retested in subsequent cycle with new integrations. This helps gradually eliminate the critical bugs from the system.

5. B. Time to market is low

In the product based companies, the competition is very high if there are too many competitors in the market. Each company tends to deliver the new product every month or week or even a day. One such example is the mobile companies where new models are release every week or two. The most important success criterion is time-to-market. One who can deliver the new features first capitalizes the business. In such situations, cycle based testing are helpful. The features are distributed over various cycles and released to end customer for testing or usage immediately.

6. C. Better visibility

When using cycle based testing, managers definitely have a better visibility over the test coverage. The test coverage spans over logical cycles and can be tracked easily. With each passing cycle, the high impacting modules get covered, ensuring greater project success and thereby, reducing risk to project delivery. The testing using this technique captures attention from senior management during test reporting.

2.4 ADAPTIVE TESTING

We have discussed various techniques in individuality in the above sections. Each technique bears its own merits and de-merits and is well-suited for specific projects or business segments based on the acceptable timelines, cost and budget or the quality of delivery. Why do we need ‘Adaptive

techniques'? Let's first define what these techniques are and then we can discuss on their needs and applicability.

2.4.1 *What are adaptive techniques?*

These techniques are being referred to as "adaptive" since these are tailored versions or combinations of original techniques. They are drafted and applied to suite the specific project objectives of delivering quality or reducing the test efforts. The adaptive techniques have their own merits and applicability.

In this paper, we will discuss on the Mix-n-Match technique which is combination of – Risk Based testing, Pair wise testing and Cycle-based technique.

2.4.2 *Need of Adaptive Techniques*

There are already many reduction techniques in place. Then, why was there a need to research or explore any new techniques. We will now discuss the need of Adaptive Techniques and the reasons why adaptive techniques were introduced and how they can be applied.

We have seen that the cycle based testing fails when there are multiple repeated cycles and each cycle requires fresh data setup and configurations or environmental customization. Often, this proves to be an overhead. The initial testing efforts are wasted in fixing defects related to environment and data setups. This diverts the focus from testing and eventually delays the testing. If these overheads are not controlled in time, they may form the cause for unforeseen frustration among testers.

Adaptive techniques can be applied in such situations to control the number of cycles but still ensuring high quality. By using, adaptive techniques you can ensure that most defects are detected earlier in testing. If the adaptive technique (mix-n-match) involves cycle based testing, you can detect most likely and critical defects in first cycle itself.

The adaptive techniques are required to overcome basic limitations of Pairwise test technique. The pairwise testing ensures the effective test reduction at 2 and 3-level interactions. However, there is always a probability of defects at higher order interactions. These adaptive techniques can make effective use of other techniques (such as risk based) to identify the critical parameters or combinations that have high possibility of defects or impose high risk to the business. By ensuring that all multi-level combinations are adequately covered, adaptive techniques eliminate the risk of low coverage or not testing unidentified combinations.

Often, in a real world, the project demands are much higher. The project managers are too particular on project delivery timelines and even, they don't want to compromise on the quality aspect. The aggressive project demands cannot be fulfilled by any of the techniques in individuality. If you as a tester try to implement risk based techniques, the quality of deliverables are at stake. If you try to target the quality goal of 100% test coverage, then risk based cannot be well suited. This is often because each technique has its own limitations and as a result, a technique which is well suited for one project scenarios may not give best results in different scenarios. E.g. techniques that are deemed suitable for regression projects cannot be applied successfully for agile projects.

The project demands can be fulfilled in two ways. One way is to do a research and evolve a better technique that can fulfill all the aspects of project deliveries. The other option is to look for specific techniques that can take benefits of the individual techniques in combinations. Adaptive technique involves combining and customizing individual techniques to draw advantages of each of the individual techniques.

2.4.3 *How to apply adaptive testing?*

Before looking at adaptive technique and their application, let's first discuss on various parameters that govern a project.

Every project has certain set of parameters that govern the test methodology and strategy to be adopted. Some of the common parameters are – complexity of application, defined quality goals, business priority and impact, time to market, delivery timelines, system knowledge etc. Practically, the combinations of these parameters define the type of technique to be applied.

If a tester has deep knowledge of a system and project is a short cycle project, the risk based testing can be considered suitable. In projects, where system is huge and test coverage requirements is over 90% with no strict timelines, cycle based testing can be considered. However, in some project, the business impact is prime focus but you do not want lose on quality, which means, high coverage and low defect density. Here, individual technique cannot ensure that all objectives are met. You, as a manager, will look forward to adaptive testing that can take advantage of both risk based and cycle based testing.

We shall now discuss on how we can perform the Mix-n-Match of the individual techniques to arrive at different forms of adaptive testing. Some of the possible adaptive techniques are listed below:

Case 1: Risk Based + Pairwise Test

- Define the parameters and levels
- Identify the high impact parameters using RBT
- Apply Pairwise technique on less/ low impacting parameters

Case 2: Risk Based + Cycle Based Test

- Identify the high impact and low impact parameters
- Define multiple cycles
- Distribute the high impact and low impact functionalities over cycles

Case 3: Pairwise Test + Cycle Based Test

- Identify parameters and levels
- Apply Pairwise technique to arrive at reduced test set
- Prioritize the test cases
- Distribute the test case in various cycles – sanity, priority and retest

2.4.4 *Measuring Adaptive Technique*

One of the most difficult questions to answer is - how to measure the adaptive techniques, how to determine the test coverage and effectiveness of reduced cases derived from adaptive techniques? The answer to this is pretty vague. So far, I haven't been able to completely answer this question. It

is difficult to analyze and practically apply a mathematical formula to compute the coverage or efficiency from the adaptive techniques.

Adaptive technique is not a standard market technique that can be learnt and applied 'as it is' without any tweaks to the project. These techniques are 'adaptive' which means they can adapt to varying projects based on business needs. Some adaptive techniques may require more of Pairwise and less on risk analysis; some may just involve Cycle-based and Pairwise techniques. However, we can relatively determine the effectiveness of this technique. The easiest way to do is the comparison of results from each technique for same project. In the case study (covered at the end), you will also find the comparison of test results in terms of defects found from each of the individual techniques and that of adaptive technique.

The task is time consuming but worth spending for beginners or research professional who can spare sometime in understanding the adaptive techniques and their benefit in comparison to other techniques.

2.4.5 *Merits, Shortcomings and Challenges*

Even the adaptive techniques are also not fool proof. There are some benefits and shortcomings associated with adaptive technique as well that are listed below.

Merits of Adaptive Testing

- Can be applied to varied project scenarios or type of testing – regression, enhancement, system, integration testing etc
- Adaptive techniques are fool proof and help to find hidden defects
- Focus is on high coverage and critical business functions simultaneously
- Can be universally applied in product based releases cycles
- High probability of defect detection.

Shortcomings of Adaptive Testing

- None

Challenges of Adaptive Testing

- Tester should have good knowledge of system/business as well as market techniques (Pairwise, combinatorial etc)
- Need thorough analysis in determining how and when to apply adaptive techniques for different projects.

2.4.6 *Suitability*

The Adaptive techniques are best suited in the following situations:

1. Quarterly Product releases

To sustain the cut-throat competition, the companies tend to deliver new features or products before committed deadline. Often the focus is on time-to-market and ensuring acceptable quality of products. In such case, instead of applying pure risk based testing, adaptive techniques are more suited. With adaptive techniques the time-to-market can be ensured along with high coverage of business risks.

2. Regression testing (High coverage and risk)

Another possible situation where adaptive techniques are well-suited is when a system need to be regression tested for new feature or technological change. The impact is on all existing modules / functions and tester needs to ensure that test coverage is high with all critical impacting functionalities are exhaustively tested. The test pattern is heterogeneous, as some functions are high impacting and required thorough testing while other less impacting functions and can be minimally tested. Here, if we tend to apply the Pairwise, the results will be homogeneous set of interacting combinations. While risk based techniques can cover only high impacting areas and would eliminated the low impacting areas. Hence, the adaptive techniques are most suitable that can ensure high coverage

3 Factors driving Reduction Techniques

There are multiple test reduction techniques and methodologies deemed applicable for different projects. Each test technique has its own merits and demerits. The test technique which may be best suited in one project or release may not be suited for another release project.

Some of the common factors that govern the suitability of reduction techniques are discussed below

A. Quick Results:

Today, every company seeks to have competitive advantage over the other companies delivering similar products. The time to market the products is reduced at cost of quality. Various techniques are researched, applied and customized to ensure that products are delivered in minimum possible time with optimum quality.

B. *Low Investment*

All companies have limited budget for various projects. The 'Cost factor' is the most influencing factor for any project. It has direct impact on the project timelines, resources and technology selected. Most companies tend to reduce the investment cost on the projects. With the test reduction techniques, a tester ensure that testing can be performed with minimum possible set of cases still keeping high productivity. This helps the project owner to draw fair estimate on how to reduce cycle time and number of resources engaged for the test activity. By adopting and applying the test reduction techniques, the project investment cost is reduced drastically.

C. Better Visibility

Today, the test techniques are being researched, understood and implemented only in small niche of IT companies. Most other companies are either not knowledgeable with using the techniques or not familiar with business benefits. The growing popularity and use of test techniques brings in added visibility to the projects and companies who have successfully implemented the techniques. The companies tend to gain an advantage by sharing their experiences, knowledge and research they did in exploring existing or new techniques.

Case study

Using Adaptive Test Reduction Techniques

We discussed earlier on various market proven techniques and their applicability, benefits and limitations. Also, we touched upon the adaptive technique – its merits and demerits. Through this case study, we will discuss how to approach the adaptive techniques. We implemented one such technique successfully in the Cross Application Project at UHG. The technique is the combination of pairwise testing; Risk Based Testing and Cycle based testing.

Project Background

The project under consideration here is an enhancement project (SPRF 20336). An enhancement is generally related to defect fixes or a new feature implementation. The project involved changes in interface functionality.

Generic Steps

Before implementing the adaptive technique, we need to understand the process how it is applied. We will first discuss the high level steps required to arrive at reduced cases (for an enhancement type projects).
We will detail them in later section.
Step 1: Analyze the functionality impacted and break it into set of parameters (as in Pairwise approach).
Step 2: For each parameter identify the set of values applicable
Step 3: Define the exhaustive set of cases from the identified parameters.
Step 4: Apply risk based technique on the parameters to determine the most impacting ones
Step 5: Re-arrange the exhaustive set matrix in such a way that most impacting parameters are on left side and least impacting ones or non-impacting ones are on right side.
Step 6: Reduce the exhaustive set matrix to keep only exhaustive set of impacting parameters only.
Step 7: Apply the pairwise technique on other non-impacting parameters.
Step 8: Arrive at the reduced set of test cases based on Pairwise and Risk Based
Step 9: Assign priorities to the cases and re-arrange them in order of impact
Step 10: Segregate the cases in multiple cycles based on priorities

In a limited time frame, execution of all the exhaustive cases is not always feasible. According to study, the probability of finding error decreases with increase in test cases as more and more parameter interactions get covered.

We will start our study using the most effective reduction testing technique - Pairwise testing. In this technique, we will initially identify the parameters and the levels. Thereon, we will switch to the adaptive technique by involving risk based testing to identify impacting and non-impacting parameters. We will then combine risk-based with pairwise to perform Mix-n-Match to arrive at reduced set of cases. With the new reduced set, we will distribute the cases over multiple cycles to perform cycle based testing. The result are not surprising but we will arrive at reduced test cases

which cover the potential business risk and still giving high coverage of over 90% distributed among different cycles.

Parameter identification

The first step is the ANALYSIS. The enhancement is studied to figure out the impacting factors (aka parameters) and their combinations. The next step is identify the factor levels (or values) for each of the identified parameters

In this case, we identified a total 9 affecting parameters. For each parameter, the number of applicable levels has been listed below (last row)

Test Scenario Parameters								
Claim type	SGA applicable	Policy Type	Business type	Product	Output	EOB Format	Plan Type	Pay To
Legacy	CX	FI	UBH	HMO	EOB	ANSI	CDHP	Member
New	N/A	ASO	Non-UBH	Non HMO	MLI Letter	MN62J	N-CDHP	Provider
					835	Current		

Exhaustive set of cases

The next step is to derive the exhaustive set of cases from the identified parameters/ levels. The exhaustive set of cases can be calculated with the simple formula.

If,

P (n) - represents the parameter count

L (n) - represents the 'No of levels' for nth parameter

The exhaustive count can be derived as

EC = L (1) * L (2)*…L (n)

In this case,

The Total Exhaustive count = 1152

Applying Risk Based Testing

As discussed previously, the risk based technique involves identifying the functionalities that have highest impact on business in terms of – costs, timelines, and quality.

All the functionalities that were directly / indirectly impacted were discussed with the business stakeholders. Like in pairwise testing, the first two steps of analysis remain the same. Initially, the parameters and the levels were identified. Next, the combination of parameters and levels were analyzed to determine the combinations with maximum impacts on business.

The business criteria's that were identified in risk based testing are listed below. These are derived based on the interaction of parameters and their impact on the application interfaces.

- Legacy claim + fully Insured
- Normal claim + FI+ CX line
- Legacy claim + FI + UBH
- FI + UBH + CX line
- Normal claim + CDHP
- Legacy claim + CX+ 62J Format
- Normal claim + MLI Letter
- Normal claim + ASO
- Legacy claim + ASO + CX line
- Normal claim + ASO + UBH
- ASO + UBH + CX line
- Legacy claim + ASO + CDHP
- Normal claim + 62J Format
- Legacy claim + ASO + MLI Letter

Now, once the parameters were identified, we defined the priority for each of the individual impacting parameters. The priorities assigned to various impacting parameters are listed below. It is to be noted that the priorities are again based on cost and probability of risk from each parameter:

Claim Type	- P1
SGA Applicable	- P2
Policy Type	- P3
Business Type	- P4
Product Type	- P5

The risk factor has been identified to arrive at P1, P2 and P3 cases as follows
P1 - where Risk > 30
P2 - where Risk > 20 and Risk <= 30
P3 - where Risk <= 20

With the priorities been defined, the exhaustive matrix was re-arranged to have most impacting parameters on extreme left and least impacting on extreme right. Using the high impacting parameters, the reduced set of 32 cases was derived. These 32 combinations form the combination of most impacting parameters and cover the maximum business risk. To certain extent, it may be considered that these combinations would be able to reduce the defect injection by 80%. However, the other less impacting parameters cannot be totally ignored. The combination of other parameters, however, need not be exhaustive.

The exhaustive matrix based on high risk parameters is listed below:

Claim typ	SGA applicab	Policy Typ	Business ty	Product
P1	P2	P3	P4	P5
New	N/A	FI	UBH	N-HMO
New	N/A	FI	UBH	HMO
New	N/A	FI	Non-UBH	N-HMO
New	N/A	FI	Non-UBH	HMO
New	N/A	ASO	UBH	N-HMO
New	N/A	ASO	UBH	HMO
New	N/A	ASO	Non-UBH	N-HMO
New	N/A	ASO	Non-UBH	HMO
New	CX	FI	UBH	N-HMO
New	CX	FI	UBH	HMO
New	CX	FI	Non-UBH	N-HMO
New	CX	FI	Non-UBH	HMO
New	CX	ASO	UBH	N-HMO
New	CX	ASO	UBH	HMO
New	CX	ASO	Non-UBH	N-HMO
New	CX	ASO	Non-UBH	HMO
Legacy	N/A	FI	UBH	N-HMO
Legacy	N/A	FI	UBH	HMO
Legacy	N/A	FI	Non-UBH	N-HMO
Legacy	N/A	FI	Non-UBH	HMO
Legacy	N/A	ASO	UBH	N-HMO
Legacy	N/A	ASO	UBH	HMO
Legacy	N/A	ASO	Non-UBH	N-HMO
Legacy	N/A	ASO	Non-UBH	HMO
Legacy	CX	FI	UBH	N-HMO
Legacy	CX	FI	UBH	HMO
Legacy	CX	FI	Non-UBH	N-HMO
Legacy	CX	FI	Non-UBH	HMO
Legacy	CX	ASO	UBH	N-HMO
Legacy	CX	ASO	UBH	HMO
Legacy	CX	ASO	Non-UBH	N-HMO
Legacy	CX	ASO	Non-UBH	HMO

Applying Pairwise Technique

Before applying the Pairwise technique, let’s discuss in brief about the interaction levels

Interaction Levels

An interaction level, in simple words, can be defined as the number of interacting parameters combinations at given point of time. An 'n-level' interaction level would signify the reduction of test cases when combination of n parameters is studied at given point of time.

E.g. A reduction based on 2-level interaction would signify the minimum number of test cases arrived at considering all possible combinations of any two parameters at given point of time. Similarly a 5-level interaction would involve analyzing all possible permutations and combinations among any 5 parameters (from the entire list).

We applied the pairwise testing at 2-level interaction on remaining 4 parameters. The result was a set of 32 cases with exhaustive set of first 5 parameters using risk based technique and reduced set of 4 parameters using pairwise testing. The final set of 32 case-matrixes derived after applying risk based and Pairwise technique is as follows

5 factors at 5th order and 4 factor at second order

- The first 5 factors are combine between them full factorial,
- Each of the first factor are combined to the other four at 2 order interaction
- Each of the last four factors are combined between them at two level interaction only

32 cases

Case	Claim type	SGA applicable	Policy Type	Business typ	Produ	Outpu	EOB Form	PlanType	Pay T
1	New	N/A	ASO	UBH	HMO	835	ANSI	NCDHP	Provider
2	New	CX	FI	UBH	Non HMO	MLI Letter	62J	CDHP	Member
3	Legacy	N/A	ASO	Non-UBH	HMO	EOB	Current	CDHP	Member
4	Legacy	CX	FI	Non-UBH	Non HMO	MLI Letter	ANSI	NCDHP	Provider
5	New	CX	ASO	Non-UBH	Non HMO	835	Current	NCDHP	Member
6	Legacy	N/A	FI	Non-UBH	HMO	EOB	62J	NCDHP	Provider
7	Legacy	CX	ASO	UBH	Non HMO	EOB	ANSI	CDHP	Provider
8	New	N/A	FI	UBH	HMO	MLI Letter	Current	NCDHP	Provider
9	New	N/A	ASO	UBH	Non HMO	835	62J	CDHP	Provider
10	Legacy	CX	ASO	Non-UBH	HMO	835	ANSI	CDHP	Member
11	New	CX	FI	Non-UBH	Non HMO	EOB	ANSI	NCDHP	Provider
12	Legacy	CX	FI	Non-UBH	HMO	835	ANSI	NCDHP	Provider
13	Legacy	N/A	ASO	UBH	HMO	MLI Letter	ANSI	CDHP	Member
14	Legacy	CX	FI	UBH	HMO	835	62J	NCDHP	Provider
15	New	N/A	ASO	Non-UBH	Non HMO	MLI Letter	ANSI	NCDHP	Provider
16	Legacy	N/A	FI	Non-UBH	Non HMO	835	ANSI	NCDHP	Member
17	New	N/A	FI	Non-UBH	HMO	EOB	Current	CDHP	Provider
18	Legacy	N/A	FI	UBH	Non HMO	835	Current	NCDHP	Provider
19	Legacy	CX	ASO	UBH	HMO	835	ANSI	NCDHP	Provider
20	New	CX	ASO	Non-UBH	HMO	MLI Letter	Current	NCDHP	Provider
21	Legacy	N/A	ASO	Non-UBH	Non HMO	835	ANSI	CDHP	Provider
22	New	CX	FI	UBH	HMO	835	ANSI	NCDHP	Provider
23	Legacy	N/A	ASO	UBH	Non HMO	MLI Letter	62J	CDHP	Member
24	New	N/A	FI	UBH	Non HMO	MLI Letter	Current	NCDHP	Provider
25	New	CX	ASO	UBH	Non HMO	MLI Letter	Current	NCDHP	Provider
26	New	N/A	ASO	Non-UBH	HMO	EOB	Current	NCDHP	Provider
27	New	CX	FI	Non-UBH	HMO	835	Current	CDHP	Provider
28	Legacy	CX	FI	UBH	Non HMO	MLI Letter	62J	NCDHP	Provider
29	Legacy	CX	ASO	Non-UBH	Non HMO	835	62J	NCDHP	Provider
30	Legacy	N/A	FI	UBH	HMO	835	ANSI	NCDHP	Provider
31	New	N/A	FI	Non-UBH	Non HMO	EOB	ANSI	CDHP	Provider
32	New	CX	ASO	UBH	HMO	EOB	ANSI	NCDHP	Member

The next step is to calculate the priorities for each case in the case matrix. The cost and risk probability associated with entire combination (this time considering all the parameters) was re-calculated and risk was determined. The risk based Pairwise set (as I call it) is defined below:

Claim type	SGA applicable	Policy Type	Business type	Product	Output	EOB Format	Plan Type	Pay To	Probability	Cost	Risk
New	N/A	FI	Non-UBH	N-HMO	EOB	Current	N-CDHP	Member	5	4	20
New	N/A	ASO	Non-UBH	N-HMO	EOB	Current	N-CDHP	Provider	5	4	20
New	CX	FI	Non-UBH	N-HMO	EOB	Current	CDHP	Provider	7	4	28
New	CX	ASO	Non-UBH	N-HMO	EOB	Current	N-CDHP	Member	7	4	28
New	N/A	FI	UBH	N-HMO	EOB	Current	N-CDHP	Member	5	5	25
New	N/A	ASO	UBH	N-HMO	EOB	Current	N-CDHP	Provider	5	5	25
New	CX	FI	UBH	N-HMO	EOB	Current	N-CDHP	Member	8	5	40
New	CX	ASO	UBH	N-HMO	EOB	Current	N-CDHP	Provider	8	5	40
New	N/A	FI	Non-UBH	N-HMO	EOB	ANSI	CDHP	Provider	5	4	20
New	N/A	ASO	Non-UBH	N-HMO	EOB	Current	CDHP	Member	5	4	20
New	N/A	FI	Non-UBH	N-HMO	EOB	MN62J	CDHP	Provider	8	4	32
New	N/A	ASO	Non-UBH	N-HMO	EOB	MN62J	N-CDHP	Member	8	4	32
New	N/A	FI	Non-UBH	N-HMO	MLI Letters	N/A	N-CDHP	Provider	6	4	24
New	N/A	ASO	Non-UBH	N-HMO	MLI Letters	N/A	N-CDHP	Member	6	4	24
Legacy	N/A	FI	Non-UBH	N-HMO	EOB	Current	N-CDHP	Member	6	4	24
Legacy	N/A	ASO	Non-UBH	N-HMO	EOB	ANSI	N-CDHP	Provider	6	4	24
New	N/A	ASO	UBH	N-HMO	835	N/A	CDHP	Member	5	4	20
New	N/A	FI	Non-UBH	N-HMO	835	N/A	N-CDHP	Member	5	4	20
New	CX	ASO	UBH	N-HMO	MLI Letters	N/A	N-CDHP	Provider	5	3	15
New	N/A	FI	UBH	HMO	EOB	Current	N-CDHP	Provider	5	3	15
New	CX	FI	UBH	N-HMO	MLI Letters	Current	N-CDHP	Member	8	5	40
New	N/A	ASO	Non-UBH	N-HMO	835	N/A	N-CDHP	Provider	8	5	40
New	N/A	ASO	Non-UBH	HMO	MLI Letters	N/A	CDHP	Member	6	4	24
New	CX	FI	Non-UBH	N-HMO	MLI Letters	N/A	N-CDHP	Member	6	4	24
New	CX	ASO	Non-UBH	N-HMO	EOB	62J	CDHP	Provider	6	5	30
New	N/A	FI	UBH	N-HMO	MLI Letters	N/A	CDHP	Member	6	5	30
Legacy	N/A	FI	UBH	HMO	EOB	ANSI	CDHP	Member	7	3	21
New	N/A	ASO	UBH	N-HMO	MLI Letters	N/A	CDHP	Provider	7	3	21
New	N/A	ASO	UBH	HMO	EOB	ANSI	N-CDHP	Provider	7	3	21
New	N/A	ASO	UBH	N-HMO	EOB	Current	N-CDHP	Member	7	5	35

An important thing to note here is that, there might still be a possibility that some of the combinations are yet not covered in the matrix. These combinations could be between the fixed (5 parameters) and reduced (4 parameters). Or, in some situations all the levels of parameters need not be considered for possible reduction. The test set generated in such cases are unnecessarily high because of interaction of one level with other parameter levels. This would eventually lead to more of testing efforts with less productivity.

It is certain that these unique/ specific combinations are not high impacting combinations but definitely affect the test coverage. In these situations, the best way is to identify and cover these combinations separately rather than re-applying the techniques. This often is judgmental and any experience tester can easily make out such unique combinations.

The information below is captured to give you a fair comparison of how the Pairwise testing (when applied individually) resulted in the reduced set at varying interaction level. We can see that as the interaction level increase the number of cases significantly increases.

Pilot using Pairwise Testing

5 order (as in 138 and 115 case scenario)
4 order (as in 64 case scenario)
3 order (as in the 32 case)
2 order (as in the 32 case)

Scenario	5 first block	Between blocks	4 last block	Num Cases
1	9	9	9	1152
2	5	5	5	138
3	5	5	2	115
4	5	?	?	96
5	5	4	4	64
6	5	3	3	32
7	5	2	2	32
8	4	4	4	65
9	3	3	3	28
10	2	2	2	13

The final step is to apply the ***cycle based testing*** technique which I consider is fairly a simple technique to apply if the priorities have been rightly defined.

A general criterion to apply a cycle based testing is to identify all the high and medium priority cases in one Cycle and low priorities in subsequent cycle. In complex projects every cycle has a pre-defined sanity testing conducted to ensure that all the basic functionalities are working as desired. The sanity cases are either judgmental, experience-based or they can be derived using Pairwise testing (especially in projects which have huge sets of cases)

Comparison of Results from Risk Based, Pairwise Testing and Adaptive Testing
We analyzed and executed the reduces sets obtained from both the techniques separately

Pairwise Testing @ 2 level interaction : 13 cases
Risk Based Testing : 32 cases
Adaptive Testing : 32 cases

Results: Risk Based Testing

	JD's Cases (32)			Defect - Risk Based Testing									
	S.No	Claim typ	SGA apply	Policy Type	Business type	Product	Output	EOB Forma	Plan Type	Pay To	Defects	Status
EOB004	4	New	CX	ASO	Non-UBH	N-HMO	EOB	Current	N-CDHP	Member	#5943	Passed
EOB005	5	New	N/A	FI	UBH	N-HMO	EOB	Current	N-CDHP	Member	#5117, #5583	Passed
EOB007	7	New	CX	FI	UBH	N-HMO	EOB	Current	N-CDHP	Member	#5117, #5583	Passed
EOB008	8	New	CX	ASO	UBH	N-HMO	EOB	Current	N-CDHP	Provider	#5064	Passed
EOB011	11	New	N/A	FI	Non-UBH	N-HMO	EOB	MN62J	CDHP	Provider	#5583	Passed
MLI003	17	New	CX	ASO	UBH	N-HMO	MLI Letters	N/A	N-CDHP	Provider	#5064	Passed
HMO001	18	New	N/A	FI	UBH	HMO	EOB	Current	N-CDHP	Provider	#5989	Passed
EOB13	23	New	CX	ASO	Non-UBH	N-HMO	EOB	62J	CDHP	Provider	#5943	Passed
MLI005	24	New	N/A	FI	UBH	N-HMO	MLI Letters	N/A	CDHP	Member	#5735	Passed
MLI006	25	New	N/A	ASO	UBH	N-HMO	MLI Letters	N/A	CDHP	Provider	#5735	Passed
NEG001	31	New	N/A	Not found	UBH	N-HMO	EOB	Current	N-CDHP	Member	#6068, #6155	Failed

Results: Pairwise Testing

Esteban's Cases (13)			Defect - Pairwise Testing								
Cases	Claim typ	SGA apply	Policy Type	Business type	Produc	Output	EOB Forma	Plan Typ	Pay T	Defects	Status
1	New	N/A	FI	UBH	Non_HMO	EOB	Current	CDHP	Member	#5117 and #5583	Passed
3	New	CX	ASO	Non-UBH	Non_HMO	835	ANSI	CDHP	Member	#5943	Passed
5	New	N/A	FI	UBH	Non_HMO	MLI Letter	MN62J	N-CDHP	Member	#5735	Passed
8	New	CX	ASO	Non-UBH	HMO	835	MN62J	CDHP	Provider	#6156	Failed
12	New	CX	FI	UBH	Non_HMO	EOB	MN62J	CDHP	Provider	#5117, #5583	Passed

Results: Adaptive Testing

Cases	Claim type	SGA apply	Policy Type	Business type	Product	Output	EOB Format	Plan Type	Pay To	Defects	Status
1	New	N/A	FI	UBH	Non_HMO	EOB	Current	CDHP	Member	#5117 and #5583	Passed
2	New	CX	ASO	Non-UBH	Non_HMO	835	ANSI	CDHP	Member	#5943	Passed
3	New	N/A	FI	UBH	Non_HMO	MLI Letter	MN62J	N-CDHP	Member	#5735	Passed
4	New	CX	ASO	Non-UBH	HMO	835	MN62J	CDHP	Provider	#6156	Failed
5	New	CX	FI	UBH	Non_HMO	EOB	MN62J	CDHP	Provider	#5117, #5583	Passed
6	New	CX	ASO	Non-UBH	N-HMO	EOB	Current	N-CDHP	Member	#5943	Passed
7	New	N/A	FI	UBH	N-HMO	EOB	Current	N-CDHP	Member	#5117, #5583	Passed
9	New	CX	ASO	UBH	N-HMO	EOB	Current	N-CDHP	Provider	#5064	Passed
10	New	N/A	FI	Non-UBH	N-HMO	EOB	MN62J	CDHP	Provider	#5583	Passed
11	New	CX	ASO	UBH	N-HMO	MLI Letters	N/A	N-CDHP	Provider	#5064	Passed
12	New	N/A	FI	UBH	HMO	EOB	Current	N-CDHP	Provider	#5989	Passed
13	New	CX	ASO	Non-UBH	N-HMO	EOB	62J	CDHP	Provider	#5943	Passed
14	New	N/A	FI	UBH	N-HMO	MLI Letters	N/A	CDHP	Member	#5735	Passed
16	New	N/A	bund on C	UBH	N-HMO	EOB	Current	N-CDHP	Member	#6068, #6155	Failed

Implementation of Adaptive Technique (Mix-n-Match)

The results obtained from each of the techniques in separation were exciting. However, we see that none of the techniques could adequately discover all the defects. Some defects were unique to Pairwise testing while others were unique to Risk Based testing.

Where is the gap?

One of the common reasons why these techniques failed in individuality is that each technique had its own way of forming combination of parameters. The combination of parameters and levels is different for both the techniques. The output test set from these techniques was not the exhaustive set, rather highly reduced set and as a result, certain interaction were eliminated which resulted in probable defect.

The combination may not be of high priority (P1) combinations but still were the potential areas for defect. These interaction are not strictly 2-level or 3-level, rather these can occur at any level interaction. With the increase in level of interaction, the probability of finding defect reduces. In other words, the defects found at higher level interaction, in most cases are low priority defects (however, this is not always true but may vary from project to project).

References

[1] Rex Black, Managing the Testing Process, John Wiley, 2002
[2] James Bach : 1999 : Software Testing and Quality Engineering Magazine, 11/99
[3] James Bach : Heuristic Risk-Based Testing :
[4] Yanping Chen, Robert L. Probert, University of Ottawa, Ontario, Canada : A Risk-based Regression Test Selection Strategy
[5] Felix Redmill in „Professional Tester“, April 2003
[6] Tom DeMarco and Tim Lister, "Waltzing with Bears: Managing Riskon Software Projects”, 2003
[7] Gretchen Henrich, LexisNexis, Product Certification & CPM : Pair-wise Testing:Moving From Theory to Practice
[8] John D. McGregor, and David A. Sykes, A Practical Guide to Testing Object-Oriented Software, Addison Wesley Inc., 2001.
[9] Schaefer, Hans: „Strategies for Prioritizing Test“, STAR WEST 1998
[10] IEEE Standard 1044-2002: Standard Classification for Software Anomalies