A Risk-Based Testing (RBT) Approach for the Masses

Bruce Kovalsky, Capgemini

bruce@kovalsky.com

Abstract

Wikipedia defines Risk-Based Testing (RBT) as "a type of software testing that prioritizes the tests of features and functions based on the risk of their failure - a function of their importance and likelihood or impact of failure". For purposes of this paper, risk related to testing can be identified as the probability that an undetected software defect from a test case will have a negative impact on the user of a system.

Testing teams have a finite amount of time (usually time-boxed) to run their tests. When crafting a strategy of how to address testing risk, organizations should test for the biggest risks first and more often, whether it is a business or technical risk. In other words, what is the minimum testing effort that one should invest in order to maximize risk reduction? It is often difficult to make informed decisions on which tests in a large set of tests planned for a test pass should be focused on -- or should not be focused on.

This paper will describe an easy-to-use process to gather risk-based information about your tests and how to use straightforward metrics in Excel to help quantitatively identify which specific tests should be run earlier and more often, and which tests should not be.

Biography

Bruce Kovalsky, PMP, *has been a Quality Assurance/Test Manager and Automation expert at various Seattle-area companies since 1990, and has been a Testing Consultant since 2002, currently with Capgemini. After receiving a Bachelor's degree in Computer Science from the University of California at Berkeley, he spent eight years developing software for the Aerospace industry, and then began focusing his career on Quality Assurance and Testing. He has presented papers at several quality conferences over the past 15 years, including the Quality Assurance Institute Conference (1998), Rational User Conference (2000, 2005), and PNSQC (1998, 2011).*

1 Why Risk-Based Testing (RBT) is important

Wikipedia defines Risk-Based Testing (RBT) as “a type of software testing that prioritizes the tests of features and functions based on the risk of their failure - a function of their importance and likelihood or impact of failure”. For purposes of this paper, risk related to testing can be identified as the probability that an undetected software defect from running a test case or test set will have a negative impact on the user of a system.

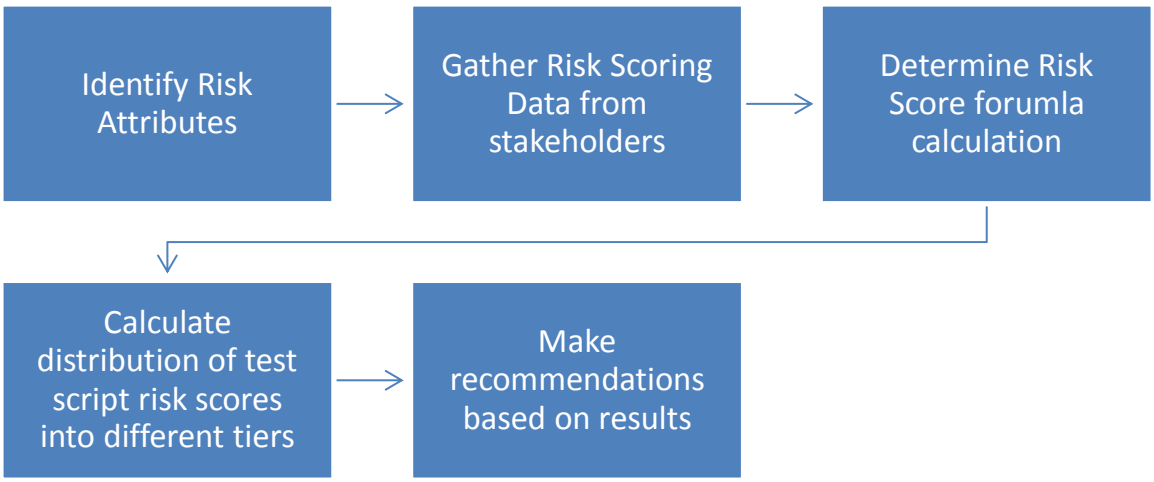
Testing teams have a finite amount of time (usually time-boxed) to run their tests. When crafting a strategy of how to address testing risk, organizations should test for the largest risks first and more often, whether it is a business, technical or other type of risk that is important to their organization.

This question is often asked of testing organizations: What is the minimum amount of testing effort that one should invest in order to maximize risk reduction? In order make informed decisions on which tests planned for a test pass should be focused on, you need an organized approach that will result in allowing your team to making these decisions, which is covered in the following sections.

2 Follow an organized approach to risk based testing

In order to properly assess risk of your testing, following this organized step-by-step approach will allow you to easily make risk-based decisions about your testing:

Figure 1: Risk Based Testing approach steps



2.1 Identify Risk Attributes

The first step is to identify the risk-based scoring attributes that will collectively be used to create a “Risk Score”. The Risk Score will be a sum of individual attribute scores, weighted appropriately as the

Copies may not be made or distributed for commercial use

organization desires; the calculation options for the risk score formula options will be covered in section 2.3.

Examples of types of risk attributes can be across a variety of dimensions:

2.1.1 Business

- Impact or criticality of a subsystem, function or feature failing (business cost of failure)
- High usage or frequency of a subsystem, function or feature

2.1.2 Technical

- Technical impact of failure of a subsystem, function or feature
- Technical probability of failure of a subsystem, function or feature
- Geographic distribution of development team

2.1.3 External Factors

- Sponsor or executive preferences / biases
- Regulatory requirements

2.1.4 E-Business / Failure-Mode Related

- Static content defects
- Web page integration defects
- Functional behavior-related failure
- Service (Availability and Performance) related failure
- Usability and Accessibility-related failure
- Security vulnerability
- Large Scale Integration failure

Typically, you should pick a mixture of attributes that will make up the Risk Score spread across multiple dimensions that are important to the company, so that each test demonstrates risk impacts across a variety of business, technical or other fronts.

For scoring purposes, a simple 1 to 3 scoring scale should be used for each attribute, where 1 = Low, 2 = Medium, 3 = High, (similar to how testers rate defect severity). You could expand this to a 1 to 4 or 1 to 5 scale, but since the risk formulas will be numerically based on the values and weighting you use for attribute scores, it's important to be consistent on how you apply the scoring values.

Once these attributes are chosen, you should define the criteria that each stakeholder group will use to rate each test attribute so they have a guide for how to score each test (see Figure 2).

Figure 2: Sample Attribute Ratings and Criteria

Business Impact Levels			Business Frequency Levels	
Rates the business impact of test failure			**Rates the frequency of test usage from a business perspective**	
3 - High	Test impacts that mean failure of the test will have a high impact to the Business. Examples:		3 - High	Very often (daily or many times per day)
	• System crash or lock-up would severely impact the user		2 - Medium	Moderate frequency
	• Usage cannot continue effectively and there is no workaround		1 - Low	Infrequently (monthly or less)
	• Inaccurate, missing or corrupt data could be used or applied that cannot be corrected			
	• Incurs financial or manufacturing costs, or scheduling related issues			
	• Non-conformance to product standards (business rules, manufacturing standards)			
2 - Medium	Test impacts that mean failure of the test will have a moderate impact to the Business. Examples:			
	• Inaccurate, missing or corrupt data could be used or applied that can be corrected			
	• Usage can continue with some workaround but it is difficult or inconvenient			
	• Moderate impact to system usability			
1 - Low	Test impacts that mean failure of the test will have a minor impact to the Business. Examples:			
	• Standard SAP transactions and processing			
	• Minor usability issues			
	• Workarounds exist or are readily available			

As figure 2 shows, you should give criteria guidelines to those who will be scoring each attribute on what each number 1 to 3 means for the specific attribute. This will provide a guide for how each stakeholder will give ratings for the attributes that they will be scoring.

Any number of attributes could be used for RBT scoring purposes, since a formula will be used that weights each of the attributes chosen to arrive at a risk score. Since variety across multiple risk categories is desired, I recommend choosing a total of four attributes, across Business, Technical or other dimensions. For purposes of this paper, I chose the following four attributes to calculate the Risk Score, two in each group:

Business	**Technical**
Business Impact	**Technical Complexity**
Business Frequency	**Technical Impact**

2.2 Gather Risk Scoring Data from stakeholders

The next step is to ask the appropriate stakeholders to score each attribute you defined for all tests in your test set according to the scoring system and criteria you have established.

A simple method for this is to extract all of you test names out of your test case management tool (e.g. HP Quality Center/ALM or IBM Rational TestManager), into Excel, and then add the attributes that you choose to include for scoring. Send the spreadsheet with blank values for them to fill out independently (see Figure 3a).

Figure 3a: Sample Attribute Scoring Data - Before

Folder	Test ID	Test Name	Technology Scorer	Technology Impact Score	Technology Probability Score
BPM	1590	048 Assign Resources to Task & Initiate Workflow	Sunil Gupta		
BPM	1690	048 Worflow and Execution Report	Sunil Gupta		
BPM	1656	048- Manager Dashboard	Sunil Gupta		
BPM	1654	048- Partial Assignment of Resource to Task	Sunil Gupta		
BPM	1850	133 Assign Resources to Projects using COE Workflow- Pool Development	Sunil Gupta		
BPM	1847	133 Assign Resources to Projects using COE Workflow-Special Projects	Sunil Gupta		
BPM	1944	148 Change Resources and Dates Post Workflow Initiation	Sunil Gupta		
BRF.	3921	01 BRF Test Finishings- HS, DC, BR, EM	Sunil Gupta		
BRF.	3913	02 BRF Test Finsihings- SS, Thermo, Flocking, Letterpress	Sunil Gupta		
BRF.	3914	03 BRF Test Finishings - Laser Cut, Stitching, Epoxy, Microbeads, Misc.	Sunil Gupta		

See Figure 3b for a sample of a spreadsheet that was filled in for the two technology scores by the developer that was asked, after receiving the spreadsheet in Figure 3a.

Figure 3b: Sample Attribute Scoring Data - After

Folder	Test ID	Test Name	Technology Scorer	Technology Impact Score	Technology Probability Score
BPM	1590	048 Assign Resources to Task & Initiate Workflow	Sunil Gupta	2	2
BPM	1690	048 Worflow and Execution Report	Sunil Gupta	3	1
BPM	1656	048- Manager Dashboard	Sunil Gupta	3	1
BPM	1654	048- Partial Assignment of Resource to Task	Sunil Gupta	2	1
BPM	1850	133 Assign Resources to Projects using COE Workflow- Pool Development	Sunil Gupta	3	3
BPM	1847	133 Assign Resources to Projects using COE Workflow-Special Projects	Sunil Gupta	3	3
BPM	1944	148 Change Resources and Dates Post Workflow Initiation	Sunil Gupta	2	2
BRF.	3921	01 BRF Test Finishings- HS, DC, BR, EM	Sunil Gupta	1	1
BRF.	3913	02 BRF Test Finsihings- SS, Thermo, Flocking, Letterpress	Sunil Gupta	1	2
BRF.	3914	03 BRF Test Finishings - Laser Cut, Stitching, Epoxy, Microbeads, Misc.	Sunil Gupta	2	3

If you have many different stakeholders that are responsible or SME's for different functional areas, a good practice is to separate each set of tests according to area of expertise in scoring when sending them the tests. For example, there could be different developers that score test cases based on the functional areas that they are developing or fixing defects for, in which case you would send spreadsheets to each developer only with test rows only that they are responsible for filling in.

As each group of risk attribute scorers enters their scores for the attributes you have defined and sends them back to you, roll them all up into a single Excel spreadsheet that collects all risk scores for all tests and all attributes.

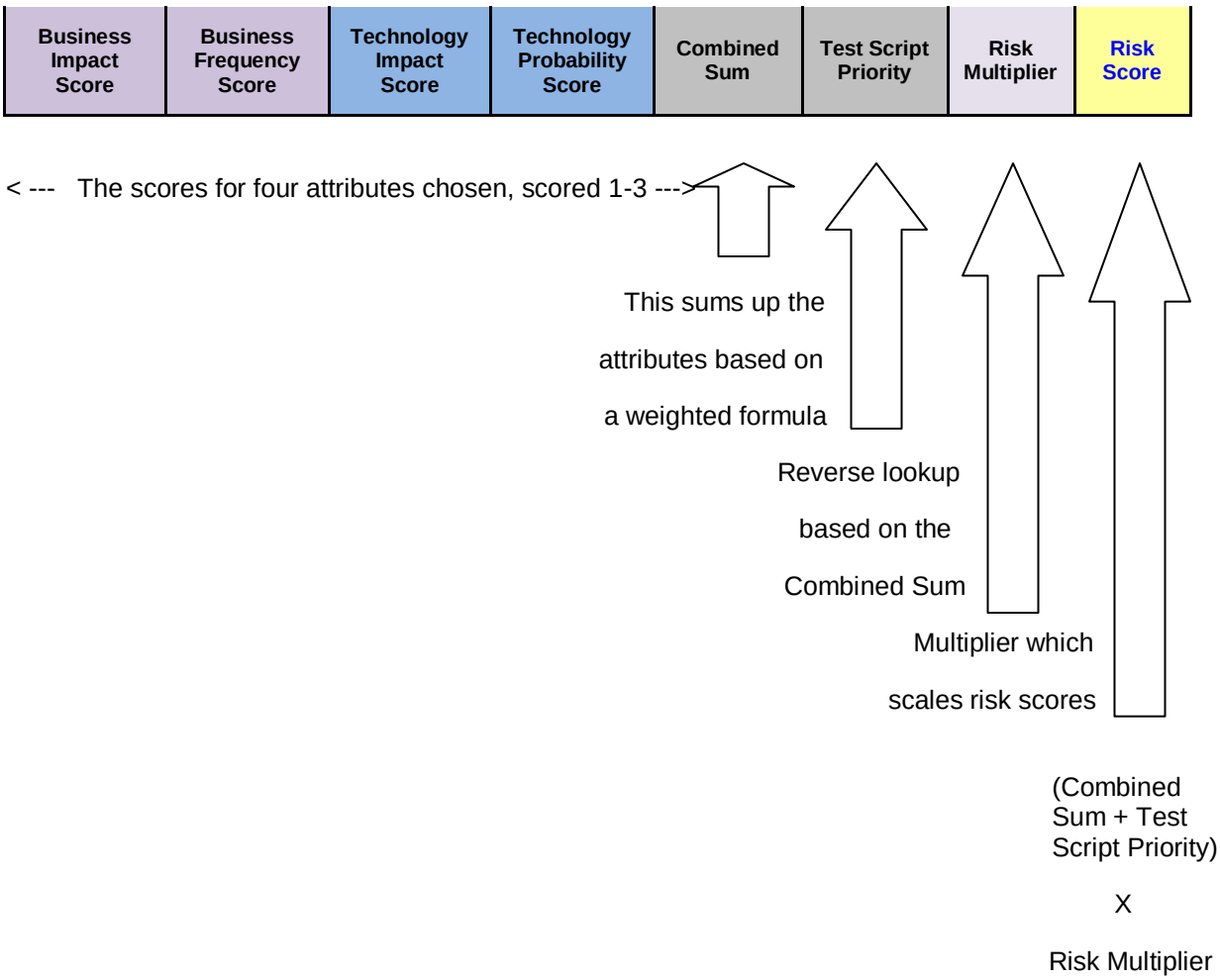
2.3 Determine Risk Score formula calculation

At this point, you have gathered raw attribute scoring data for each attribute that you have defined initially, for all test cases. Next, determine the appropriate formula to be used to calculate your risk score.

The ideal technique is for the total collection of Risk Scores for each test to be distributed over a range of values, so that they can be fit into High, Medium and Low tiers that will allow you to make decisions on which tests are High tier vs. Low Tier.

The Risk Score is based on the following variables shown in figure 4 below:

Figure 4: Makeup of the Risk Score



Assuming you have four attributes chosen at the bottom of section 2.1, some different approaches to calculate the Combined Sum formula (which directly drives the Risk Score formula) are:

2.3.1 Evenly weight the attributes

For this method, you simply weight each of the attributes the same, and sum up the values. So the Combined Sum formula would be:

= (Business Impact + Business Frequency)/2 + (Technology Impact + Technology Probability)/2

2.3.2 Weight one or more attributes higher than others

In this case, you would value one or more attributes some percentage higher than others so that scores for that attribute would have more weight that the others when the final Risk Score is calculated. For example, let's say the business team feels that the Business Impact score should have 50% more weight than the Business Frequency score, and keep the Technology scores the same weight. In this case, the Combined Sum formula would be:

= ((1.5*Business Impact) + Business Frequency)/2 + (Technology Impact + Technology Probability)/2

Note in the above case, by weighting Business Impact by 1.5 times the Business Frequency Score, the other mathematical impact to the Risk Score is that it gives both of the Business Scores a 25% boost over the Technology Scores.

2.3.3 Make one attribute override all other attributes

In this case, you would value one attribute over all others. For example, if someone scores Business Impact Score as a High (= 3), then the overall Risk Score should be the highest possible score and tier, otherwise, they would all be weighted evenly as in the first case above. For this case, the Risk Score formula would be:

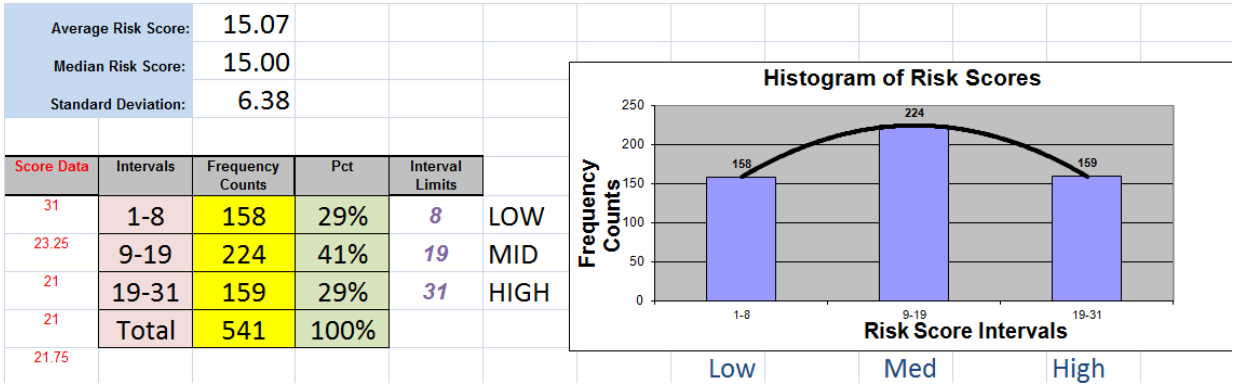
=IF (Business Impact=3, <Highest Score>, SUM (Combined Sum + Test Script Priority) * (Risk Multiplier))

Other variations on the formulas are possible, but the above three are all suggested ways you can weight your formula to arrive at the optimal Risk Score distribution for you organization.

2.4 Calculate distribution of test script risk scores across different tiers

After all the initial scoring data has been collected, use a spreadsheet with formulas to calculate the distribution of the risk scores into different tiers (e.g., High, Medium, Low) which can be set by changing the interval limits that the risk scores are distributed over (see Figure 5).

Figure 5: Distribution of Risk Scores into Tiers and Histogram



By using filters in Excel, you should review how all of the Risk Score data distributes based on the formula used, and set the Interval Limits appropriately to allow the data to fall into different tiers.

The formula to use for the Frequency Counts, which counts the number of tests that fall into the interval range specified, is the following:

= FREQUENCY (Score start range : Score end range, Interval Limit start : Interval Limit end)

By observing the distribution of risk scores, you can modify the Interval Limits to create different intervals, to obtain the desired tier distribution. Optimally, the distribution should try to emulate a Normal distribution (Bell Curve-like), but any rough approximation of this shape is acceptable. The important theme is you don't want too many tests falling in the High Tier, since your organization can only focus on so many high risk tests. By changing the formulas in section 2.4, and adjustment the tier intervals, you can achieve the best possible tier distribution, as desired.

Another useful method is to show tables on how each attribute was scored individually, so you can observe the raw data of how the Risk Scores were derived; see Figure 6 for a sample of how this is gathered.

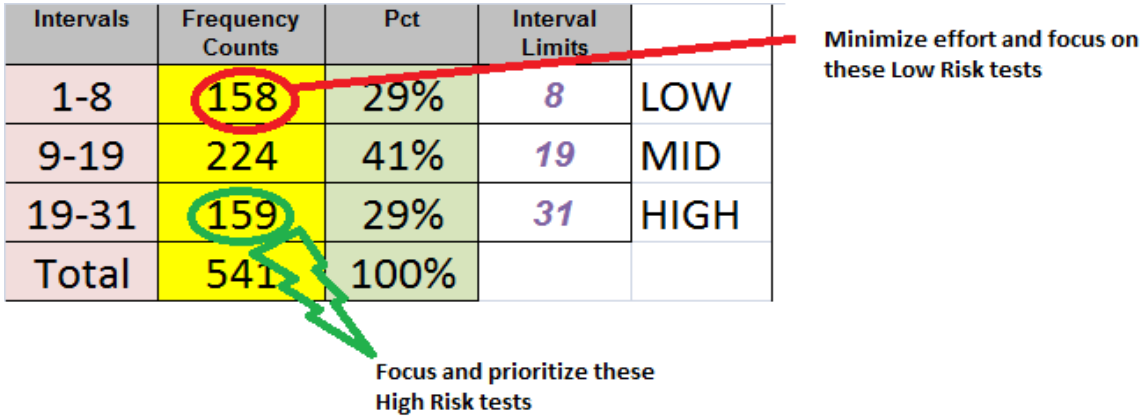
Figure 6: Separate Attribute scoring data tables

	Business Impact Score	%	Business Frequency Score	%	Technology Impact Score	%	Technology Probability Score	%
3-High	257	48%	183	34%	72	13%	57	11%
2-Med	200	37%	214	40%	253	47%	224	41%
1-Low	84	16%	144	27%	216	40%	260	48%
	541		541		541		541	

2.5 Make recommendations based on results

Once you have finalized the Risk Score formula and have a Tier Distribution that is accurate, decisions can then be made on which tests should be run earlier and more often (High Tier tests), and which ones should not be focused on (Low Tier tests), as shown in Figure 7.

Figure 7: Risk Score Distribution focus



As the figure emphasizes, use the results of how the risk scores were distributed into the High, Medium and Low tiers to determine which tests to focus on, and not focus on for your testing efforts going forward.

3 Risk Based Testing Approach Summary

Software testing as an exercise helps answer the basic question of when an organization should release their software to production at the highest quality with the least amount of risk. At some point in your testing career, management will inevitably ask your testing organization the question:

What is the minimum amount of testing effort that we can do now in order to maximize risk reduction?

By following the Risk Based Testing steps described in section 2 in an organized process, you will be able to easily quantify and categorize any set of existing tests to be run across easily identifiable risk tiers, according to the customizable testing risk attributes your organization chooses

Once your tests have been broken down into High, Medium and Low risk tiers, the decisions to make of what to test first, with more resources, and more often becomes simplified.

4 References

Web Sites:

Wikipedia definition of Risk Based Testing:

http://en.wikipedia.org/wiki/Risk-based_Testing

Wikipedia definition of Normal distribution:

http://en.wikipedia.org/wiki/Normal_distribution

Andy Tinkham, Modern Apps Live (MAL), QAT and Automated Testing of Modern Apps

http://www.slideshare.net/andytinkham/mal12-qa-tandautomatedtesting

Stephane Besson, A Strategy for Risk Based Testing

http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=ART&ObjectId=7566&tth=DYN&tt=siteemail&iDyn=2

Sample RBT Excel Spreadsheets:

GoogleDocs link to PNSQC folder

Some sample RBT spreadsheets will be uploaded for attendees to download and use for their efforts using different formulas I cover during the presentation.

Copies may not be made or distributed for commercial use