ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 8, Issue, 03, pp.27918-27923, March, 2016

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

RISK ANALYZER - CALCULATION METHODOLOGY

***Kaushik Halder**

Employee of Accenture, Pursuing FPM from National School of Leadership in Marketing Research and Analytics, Pune – 412105, India

ARTICLE INFO

Article History:
Received 26th December, 2015
Received in revised form
15th January, 2016
Accepted 08th February, 2016
Published online 16th March, 2016

Key words:

Module,
Software Testing,
NFR Testing,
Risk,
RACI model,
IT.

ABSTRACT

Usually in any project, the end Business requirement is to implement software in production flawless and for that to happen, each modules in the project should have minimum below properties for it to get integrated and automated with other modules.

i. Least or no manual intervention while execution of module in production
ii. Module uniqueness should be there and to avoid redundancy the same module should called by other modules.

This proposition is for easy maintenance. Module redundancy is hard to propagate as the same is costly affair during maintenance. This is mainly design approach. Every time the above issues found during software testing, the same were communicated to stakeholder usually in qualitative form. The issue and impact explanation in qualitative form is tougher because one has to explain the module importance based on which issue priority will be determined. Stakeholders are also liable to understand the domain of module to understand it's importance and it's impact factor during failure. The whole process becomes lengthy in acknowledging to issue. It is believed if whole communication would have taken place in quantitative form, then time taken to respond to issue would have been shortened. Taking an scenario which had happened with author in real time, Author will try to share fact how quantitative factors will help to resolve faster in compare to qualitative factors. Once, Author was surprised to found his salary credited was almost halved. Author was stressed of the fact and tried explaining his supervisor and payroll people the importance it carries as he was sole earning member and also not carrying that much amount in bank to address requirements. He went on explaining the impact of not getting full salary by addressing to fact that his corporate card payment will be on hold, his own personnel card payment and other utilities payment will be on hold and impact went on. Getting hold of the nature and gravity of issue, payroll department decided to look into his issue with highest priority. After everything got resolved, Author tried analyzing the situation and felt bad for bringing his personnel stuffs to office. He felt, it could have been explained in numbers by saying, out of scale of 100 the importance the issue carries was 98 and impact factor was 10000. Which says high importance and non zero impact. The quantitative or number system would have been easier from Author perspective to explain and would have been better for other stakeholders to respond. Time taken to respond to issue would have been minimized.

Citation: Kaushik Halder, 2016. "Risk analyzer - calculation methodology", *International Journal of Current Research,* 8, (03), 27918-27923.

INTRODUCTION

In one of Organizational internal engagement, Author found modules before getting into production were having following properties

i. For each module to execute manual intervention was required.
ii. Module Complexity: This is actually the risk, which is calculated from end user point of view. Ideally the module should have 0 score for module complexity for getting it implemented in production. More the module complexity score more is risk in implementing in production.

This document is focused on the method of calculating Importance and Complexity of a sub Application / Module in a whole Application. From Software testing perspective, this is a part of NFR Testing or in other words this can also be part of Black box or nonstructural testing phase. Author devised a mathematical process to calculate module Importance to eradicate irrelevant and redundant Module / Application. Ideally more the value, better is Architecture to define Module / Application. Maximum value it can achieve is 2 (out of

**Corresponding author: Kaushik Halder,*
Employee of Accenture, Pursuing FPM from National School of Leadership in Marketing Research and Analytics, Pune – 412105, India.

scale 2). Author also devised mathematical process to calculate complexity and there is no threshold for good or bad but ideally complexity has to achieve "0" for full automation. Relatively, more the complexity number inferior is the quality. Complexity number reveals the risk associated with the module and the same is calculated based on Module importance.

Method – Calculation Methodology

Module Importance: Importance of an application is inference by the upward and downward dependency. It is quantified by calculating the score. This method is meant for testing process automation by detecting redundancies / manual intervention / reducing unnecessary process. Below template used to calculate Module Importance Score and example taken in the table describes each attributes

Explanation of Attributes: Sl.No#: Module Index number/ Serial Number

Module Name: Name of Module

List of Activity/Sub Activity: List of all activities and outputs at end of execution of modulee. g; "Test 1" module Creates an output file "A", which say it to be Activity A. After creation of file "A", partition of records are done in two parts and say the activity to Activity B. Activity B is taken care by program automatically. After Activity B is over, The salary column in 1st partition file is multiplied with 0.3 and salary attribute in 2nd partition is multiplied with 0.35 and say this activity to be Activity C. Activity C is manual.

List of Activity taken care Manually: List of activities among all activities listed under heading "**List of Activity/Sub Activity"** taken care manually. e.g; Following the activities from above example it is known only Activity C is carried out manually.

% of Non Manual Inference: Number of Activities carried out automatically in terms of percentage e.g; Following the example from above 2 Activities out 3 are automated. So, (2/3)*100 =66.66%.

Other App name, where list of activities are present: Name of the Modules/Application where same Activities are carried out e.g; Following above example, Application "Test 2" where Activity B , which is meant for portioning of file is repeated .

List of Activity present in Other Activity: List of Unique Activities found same in other Applications/Modules e.g; Following above example, Application "Test 2" where Activity B, which is meant for portioning of file is repeated.

% of non-Redundancy: It is to focus on number of Activities seemed to be non-redundant in other Modules/Applications e.g; Following above example it seems two Activities are non-redundant and one Activity is redundant. So,(2/3)*100=66.66%.

Up Stream Dependent: It is know if the Application/Module is upstream dependent.

Down Stream Dependent: It is know if the Application/Module is downstream dependent.

Module Importance Score: Formulae to calculate application/Module Importance Score. Formulae it follows as below ((% of Non Manual Inference + % of non-Redundancy)/2) * (Up Stream Dependent (value is 1 if dependent or 0 if not dependent)+Down Stream Dependent(value is 1 if dependent or 0 if not dependent)).

Explanation on achieving Formula

Module importance is referring to three dimensions as below

i. Non Manual Interference
ii. Non Redundancy
iii. Dependencies

From "Non manual Interference", it speaks about the percentage of automation that has taken place and thus it's important for a module to know in quantitative manner the left out scale to be achieved for fully automation. The "Non Redundancy" means module uniqueness. This dimension tries to quantify the uniqueness in scalar form. The "Dependency" both upstream and downstream defines the importance of activity being carried out in the particular module.

So, Module Importance is directly proportional to all three dimensions ("Non Manual Interface", "Non Redundancy" and "dependencies")

Note: Formulae is multiplication of Average of "% of non Manual Interface" and "% of non Redundancy" and summation of dependency. Average is taken, to reduce the numerical factor and to retain the impact even at least one is having more than 0%. Importance enhances with dependencies and hence Summation of dependency is multiplied with Average of percentage of "Manual Interface" and "Redundancy". The process remains same for all Applications/Modules so relative impact based on the scalar factor of complexity score remains same.

Module complexity or risk: It is to check how easy are the modules to use for end user. Ideally, Module should be easy to use for an end user (not necessarily end user will be Data Scientist, if the module is related to Machine learning). It is not going to be "Cyclomatic Complexity" test because it demands structural test and On Contrary, In an Analytical module there are various algorithm available which are unsupervised and complex in its own way. Rather, it will follow below template (which may change based on requirement) to know how much it is complex from end User View (Details of each column is described in Table itself). This method helps to know how far a module or an Application is automated. Less is the Module complexity better is the result for automation.

Explanation of Attributes: Sl. No#: Module Index number/ Serial Number

Module Name: Name of Module

Module Importance-Score: Score being carried out from calculation being made in "Module Importance" (described in above session).

No# of times User Interface occurred: It is number of User input interface (in form of text box or combo box , check box etc) or any action (clicking buttons) that User need to take for a module or an Application to execute. In Summary, It is number of time manual intervention occurred.

Knowledge– Up Stream Dependency: Does End User require Up Stream Dependency Knowledge, while executing the Module. Value is expected in the form of 1 or 0. If End User requires up stream dependency knowledge then value would be 1 else value would be 0.

Knowledge - Downstream Dependency: Does End User require Down Stream Dependency Knowledge, while Executing the Module. Value are expected in the form of 1 or 0. If End User requires downstream dependency knowledge then value would be 0 else value would be 1.

Complexity Score: Formulae to calculate Application/Module Complexity Score. Formulae it follows as below

((No# of times User Interface occurred) * (Up Stream Dependent (value is 1 if dependent or 0 if not dependent)+ Down Stream Dependent(value is 1 if dependent or 0 if not dependent))) / Module Importance-Score

Explanation on achieving Formula

Module Complexity is referring to three dimensions as below

i. Module Importance
ii. Number of times Manual Interpretation
iii. End user's knowledge on upstream and downstream Applications

Module Complexity decreases with increase of "Module Importance factor". "Module Importance factor" increases with reducing redundancies and manual interference hence it is inversely proportional to Module Complexity because it increases with increase in manual interferences. With increase in "Number of times manual Interpretation", Module Complexity increases. For an End User, the application should process without having dependency on end user's dependency knowledge.

So, Module Complexity is inversely proportionally to "Module Importance" and directly proportional to

"Number of times Manual Interpretation" and "End user's knowledge on upstream and downstream Applications".

Note : Complexity score or Risk Score is intended to find complexity of software on end user behalf rather program complexity. More over as the process remains same for all App's so relative impact based on the scalar factor of complexity score remains same.

Note: End Users refereed in the document are referring to employees/associates belonging to "R" Group in "RACI" model. The same is discussed more in Session 9.

Result: Calculation Methodology– Template

The methodology developed is a framework, which can be tailored based on business requirement. Using this methodology Module's Importance is quantified and risk for a module to be in production is also quantified. Quantifying the terms are easily understandable by all level of Users/Stakeholders. It is a scale to measure the milestone, which is scalar in nature and easy to be conveyed and make all level of Users understand of the Importance and Risk of the module. The matrix can betailored based on functional/project need.

Discussion: Use Cases

Following above frame work/methodology two use cases described below and one of them is related to software life cycle and the other one is not related software lifecycle. It is to show, the methodology developed is compatible with various scenarios.

Use Case -1 (NON IT USES CASE): Mr. X who is service holder and having family. One fine day he felt doing something new in Kitchen for his family. He is not kitchen savvy so to start with and as warm up session he felt it's good to start with "Potato Fry(French Fries)".Matrix attached to quantify how much the dish was important and complex from Mr. X point of view. Discussions/ Notes are described in excel sheet.

From the attached Matrix, Activity Importance score for Mr X is 1.33 (out of scale 2). Which means few Activities carried out are manual and not automated. Importance from Mr. X perspective could have enhanced if machine would have used to reduce manual efforts like slicing potatoes etc.

Assumption: If Mr X would have carried out same activity in MD (Mc. Donald) and incurred importance 1.33 then MD being customer centric would have definitely put effort to achieve the 2. It is so because the Employees in "R" (R in RACI model) group in MD carries vital role in facing customer and if importance reduces then manual activity enhances which in turn will reduce the quality in dealing with clients. Similarly, Complexity score for Mr. X is 64.66. This is because of huge manual intervention. In MD, manual interventions are reduced to almost ZERO. Dependency knowledge is also not required for employees in "R" (R in RACI model) group in MD because French fries carries no different options and ingredients for serving French fries are pre defined. Hence in case of MD, complexities from employees perspective in "R" of RACI model reduced to 0.

Use Case -2 (IT USE CASE): It is same Mr. X, who is working in IT firm and playing the role of Level-3 in production support. His daily job is to solve or find root cause for issues / production failures based on Ticket priority and importance. Ticket Priority and importance are aligned to Business requirement. At time Mr. X has to define Ticket importance and complexity based on his requirement to mitigate the gap of delivery.

Sl. No# (Attribute-1)	Module Name (Attribute-2)	List of Activity / Sub Activity (Expected at end of execution of each module) (Attribute-3)	List of Activity being carried out Manual (Attribute-4)	% of non-Manual Interference (Attribute-5)	Other App name , where list of activities are present (Attribute-6)	List of Activity present in Other Activity (Attribute-7)	% of non-Redundancy (based on unquie activity present in other App/ Module) (Attribute-8)	UpStream Dependent (Yes/No) (Attribute-9)	Down Stream Dependent (Yes/No) (Attribute-10)	Module Importance Score (Attribute-11)	Remarks (Attribute-12)
1	Test1	1. Activity A 2. Activity B 3. Activity C	Activity C	(2/3*100)=66.66%= 0.66	Test2	Activity B	(2/3*100)=66.66%= 0.66	Yes	No	Multiply = ((Attribute-5+Attribute-8)/2) *(Attribute-9(Values either 1 or 0) +Attribute-10(Values either 1 or 0)) =(((0.66+0.66)/2)*(1+0))=0.66	

Sl.No# (Attribute-1)	Module Name (Attribute-2)	Module Importance-Score (Attribute-3)	No# of times User Interface occurred (Attribute-4)	Dependency is required when App's are involved in providing end result as a whole		Complexity Score (Attribute-6)	Remarks (Attribute-7)
				Knowledge - UpStream Dependency (Attribute-5.1)	Knowledge - Downstream Dependency (Attribute-5.2)		
1	Test1	0.66	How many times , manually inputs are given .	Does End User require UpStream Dependency Knowledge , while Executing the Module .	Does End User require DownStream Dependency Knowledge , while Executing the Module .	Multiply : (Attribute-4* (Attribute-5.1(1 if Yes else 0)+ Attribute-5.1(1 if Yes else 0)))/Attribute-3	

Use Case I:
Module Importance Score:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Sl. No# (Attribute-1)	Module Name (Attribute-2)	List of Activity / Sub Activity (Expected at end of execution of each module) (Attribute-3)	List of Activity being carried out Manual (Attribute-4)	% of non-Manual Interference (Attribute-5)	Other App name , where list of activities are present (Attribute-6)	List of Activity present in Other Activity (Attribute-7)	% of non-Redundancy (based on unquie activity present in other App/ Module) (Attribute-8)	UpStream Dependent (Yes/No) (Attribute-9)	Down Stream Dependent (Yes/No) (Attribute-10)	Module Importance Score (Attribute-11)	Remarks (Attribute-12)
2	1	Evening Snacks	1. Collect 800 gms of potatoes 2. Wash and Slice All potatoes 3. Collect 3 cup of floor , 1 table spoon lemon juice , 1/4 table spoon black pepper , 1 table spoon salt , 1 table spoon oil and water 4. Make batter of all ingridents mentioned in point no# 2 5. Heat oil in large skillet 6. Dip potato slices one at a time in the batter and place it in hot	1. Point no#2 : Wash and Slice All potatoes 2. Point no#4 :Make batter 3. Point no#5 :Heat oil in large skillet 4. Point no#6 :Dip potato slices one at a time in the batter and place it in hot oil	(2/6*100)=33.33%= 0.33	NA	NA	(6/6*100)=100.00%= 1	Yes	Yes	Multiply = ((0.33+1)/2) *(1+1) =(((0.66))*(2))=1.33	
3	Explanation	Plan to make French Fries	After the plan is freezed , above mentioned activities exceuted	Out of 6 activities in attribute-3 , 4 activities which invited extended physical activity are only regarded as manual activity	Out of 6 activities in attribute-3 point 1, and point 3 are not considered as manual activities .	No other Activity or plan after evening snacks	No other Activity or plan which replicates any 6 activities mentioned in Attribute-3	NO Redundancy , so all 6 activities are unique	The amount of Evening Snacks depends on Lunch .e.g; More lunch less evening snacks and vice versa .	Evening Snacks has down strem dependencies on Dinner .e.g; More evening snacks then less Dinner and vice versa .	Plan of making French Fries carries Importance 1.33 out of scale 2	

Module Complexity Score:

Sl.No# (Attribute-1)	Module Name (Attribute-2)	Module Importance-Score (Attribute-3)	No# of times User Interface occurred (Attribute-4.1)	No# of times User Interface occurred (Attribute-4.2)	No# of times User Interface occurred (Attribute-4.3)	No# of times User Interface occurred (Attribute-4.4)	Dependency is required when App's are involved in providing end result as a whole		Complexity Score (Attribute-6)	Remarks (Attribute-7)
							Knowledge - UpStream Dependency (Attribute-5.1)	Knowledge - Downstream Dependency (Attribute-5.2)		
1	Evening Snacks	1.33	10 minutes	4 mins	4 mins	25 mins	1	1	Multiply : ((10+4+4+25)* ((1)+(1))) /0.33 = 64.66	
Explanation	French Fries	Module Importance calculated	10 mins taken to slice 800 gms of potatoes (approx 10 potatoes)	4 mins taken to make batter	4 mins to heat oil	25 mins to fry approx 200 sliced potatoes	It requires the knowledge on on following i. How many people are there and there tastes	It requires the knowledge on on following i. Along which other side dish / main course the French fries are served	Plan of making French Fries carries complexities 64.66 . For Complexities there is no scale and 0 expected to	
			All four manual interfaces were measured in terms of time							

Use Case II:

Module Importance Score:

Sl. No# (Attribute-1)	Module Name (Attribute-2)	List of Activity / Sub Activity (Expected at end of execution of each module) (Attribute-3)	List of Activity being carried out Manual (Attribute-4)	% of non-Manual Interference (Attribute-5)	Other App name , where list of activities are present (Attribute-6)	List of Activity present in Other Activity (Attribute-7)	% of non-Redundancy (based on unqie activity present in other App/ Module) (Attribute-8)	UpStream Dependent (Yes/No) (Attribute-9)	Down Stream Dependent (Yes/No) (Attribute-10)	Module Importance Score (Attribute-11)	Remarks (Attribute-12)
1	Ticket 102	1. Check if such warning happened in history 2. Check recent source files and check for data 3. Check for at what execution level , the warning level is comming up 4. Check for recent look ups 5. Check for schedule it got triggered 6. Check for the end result ,if it is up to expectation	All 6 points in Attribute-6	(0/6*100)=0%= 0.00	NA	NA	(6/6*100)=100.00%= 1	Yes	Yes	Multiply = ((0+1)/2) *(1+1) =(((0.5))*(2))=1	
Explanation	The Ticket is related to production warning (not failure) and is new of its kind carries low business priority and importance and SLA is 7 days	After Ticket got assigned , it's huge set of manual work that need to be executed . A special attaintion has to be given to deliver in required time . All set of actions are manual and non repeatable .	Out of 6 activities in attribute-3 , All 6 activities which invited extended manual activity .	Out of 6 activities in attribute-3 all activities are considered as manual activities .	No other Activity or plan after evening snacks	No other Activity or plan which replicates any 6 activities mentioned in Attribute-3	NO Redundancy , so all 6 activities are unique	It depends on on the processed feed received from upstream . Level-3 associate need to analyze processed file based on described actions	How much the downstream is impacted because of warning has to be analyzed . Even if there is no downstream dependent module , the end result has analyzed .	Plan of solving ticket 102 carries Importance 1 out of scale 2	

Module Complexity Score:

Sl.No# (Attribute-1)	Module Name (Attribute-2)	Module Importance-Score (Attribute-3)	No# of times User Interface occurred (Attribute-4.1)	No# of times User Interface occurred (Attribute-4.2)	No# of times User Interface occurred (Attribute-4.3)	No# of times User Interface occurred (Attribute-4.4)	No# of times User Interface occurred (Attribute-4.5)	No# of times User Interface occurred (Attribute-4.6)	Dependency is required when App's are involved in providing end result as a whole		Complexity Score (Attribute-6)	Remarks (Attribute-7)
									Knowledge - UpStream Dependency (Attribute-5.1)	Knowledge - Downstream Dependency (Attribute-5.2)		
1	Ticket 102	1	240 mins	600 mins	240 mins	240 mins	240 mins	240 mins	1	1	Multiply : ((240+600+240+240+240+240)*	
Explanation	It's a production ticket , which is raised because of getting warning status in log and not because of failure . As part of business , it carries less importance and less priority .	Module Importance calculated	240 mins to check for 1 year(since last application upgraded) history logs	600 mins to check source files and data	240 mins to reexecute to locate at what point warning message is coming up	240 mins to check and revalidate the data and data types for all look ups	240 mins to check for the scheduler and their dependencies	240 mins to check/validate for the end result	It requires the knowledge on on following i. Sorce input file being created by upstream module	It requires the knowledge on on following i. How the warning impacts the end result	For employees in R group (refering RACI model) will have complexity of 1802	
			All six manual interfaces were measured in terms of time									

Usually, his importance rises when a Ticket referring to issue is new of it's kind and demands on least/ no repetition of previous work/task because he has to devote more time to get it solved. These kind of new tickets may also invite huge manual intervention and may result in enhancing Complexity.

For Mr. X the ticket 102 is fairly new type, hence all the plan of actions are non-redundant (usually, for known issues actions / scripts to be used are known or reused and hence are redundant and not unique) so it enhances the importance of the Ticket. The complexity enhanced with huge manual effort. In such environment, this kind of matrix referring to Importance/complexity can also be plotted in graphs (in more user friendly manner) and further can be shown as complexity reducing at end of each action item. More on Automation/Integration described more in next session.

NOTE : Matrixes for both Use Cases described below.

Conclusion

The framework developed mainly keeping in persons who fall in the category "R" of "RACI" model. RACI refers to people in each four roles within a project

i. "R" represents "Responsible" and these are persons who are actually doing the work, and are expected to actively complete the tasks. These persons are "end User" from stake holder point of view and are either Developer / Tester / Level 3 etc.
ii. "A" represents "Accountable" and these persons are ultimately assure able for an activity or Decisions. They are usually Project managers/Program managers in the project .
iii. "C" represents "Consult" and these are people who are typically domain experts .They are usually consultants / Domain experts/SME etc.
iv. "I" represents "Inform" and these are the individuals who need to be informed during Project.

The framework/methodology is about scoring each event of persons falling in "R" to represent how the event is important or complex from their point of view. This is for other persons falling in group like "A" and "I" to get essence of resource capability, business and technology oriented risk, helps in taking respective strategic decisions and it is a unique way of communication between different levels of RACI model.

Summary

1. Importance is calculated out of score 2 (it may be 200 as well if calculated in percentages). More Importance defines activity/module to be more unique in nature (carries no repetitive in nature) and less manual interventions/activities.
2. Complexities has no upper boundaries but objective has to be set to achieve 0. Complexity 0 is good in terms of automation and integrity between activities or modules.
3. Depending on requirement, the frame work/methodology can be tailored and can further planned for integration with different sources for input data.

Acknowledgement

Author is much grateful to his previous Supervisor Mr. Rahul Kongbrailatpam and thankful to his family , who are always by his side during his research work .

REFERENCES

Subject RACI: http://en.wikipedia.org/wiki/Responsibility_ assignment_matrix, checked on date : 17-Feb-2016. Time 10.00 AM.

Subject RACI: http://www.valuebasedmanagement.net/ methods_raci.html, checked on date 17-Feb-2016. Time 11.31 PM

****