Models for evaluating review effectiveness

KiranKumar Marri
Project Manager
Product Competency Center
Infosys Technologies Limited, Bangalore

Abstract: *Delivering a high quality reliable product is the main focus in any software development. The basic quality measure is the defects in the product. Defects found in the later phases of the product development are mainly because of faulty design and code and poor reviewing capability. The role of the reviewer and tester are crucial to avoid these defects. How do we study, evaluate and quantify the effectiveness of reviews in general and group in particular, during design review, test plan review and test cases review? An important parameter in such studies pertains to the estimation of an individual reviewer's ability. This article proposes simple estimation framework and illustrates its potential applicability.*

Introduction: Software product development and defects co-exist in the information and software industry. Delivering a high quality reliable product is the main focus in any software development. The basic quality measure is the defects in the product. A systematic software process at different life cycle stages should be followed to deliver zero-defect high quality software. Yet, we come across many software products having defects at different stages, and sometimes even after the delivery and acceptance. Overlooked defects (like faults in the software system requirement, design or code) propagate to subsequent development phases, and detecting and correcting them becomes more and more difficult as the phase goes to completion [5]. The reliability of the delivered product has maximum impact when the root cause of the defect is because of the faulty requirements [8]. By having good software testers, it is possible to trace the defects, but at the cost of delayed schedule, slippage in delivery and additional increase in production cost. The relationship between the product quality, and process capability and maturity has been recognized as major issue in software engineering, based on the premise that improvement in process will lead to higher quality products [6].

The standard process followed in software development is requirement phase, design phase, coding phase, testing phase and delivery on acceptance. Several authors have published the process in detail [2,3,4].

What is review process? The inspection of system requirement documents is the first stage of review. During the design and coding phase, it is a normal process to review the technical documents or the source code, by a group of experts or individual. The process of checking the technical documents in these phases is called **review process**. The desirable characteristics of a high quality design document are readability, clear objectivity, minimum complexity, maximum coverage of all the possible issues, flexibility towards any future changes, alternate design comparison, unit test cases to name a few. Since the decision to finalize the next stage of implementation or coding phase depends on the design phase, the review process is very critical.

The reviewing process details are mentioned in reference [4]. A typical review process has four steps.

Step 1: The developer prepares the technical document for the design or code review. The template of the design document varies, but contains the general structure as mentioned above.

Step 2: The developer submits the technical document to the reviewers, well before the review date. The reviewer evaluates the document and tries to detect all the defects in the design/code/document. From the reviewer's viewpoint, the quality of design which is evaluated or examined, should be based on

Parameter	Description
Matching Requirements	The requirements, which are defined, should match the design or the prototype submitted.
Stability	Is the proposed design stable and reliable?
Flexibility	In the event of any change in requirements, how flexible is the design to accommodate any further change?
Alternatives	Every module designed can have an alternative solution, and whether the alternative solution has explored.
Impact	The impact of any other module design on the design under review. This is important during the integration phase.
Error handling	Whether the proposed design will handle the error conditions. How it is handled?

Step 3: The moderator and the reviewers review the technical document. The reviewers express their views on any issue in the document or the design. The developer clarifies the issues raised by the reviewers. The defects are to be identified and the risk areas noted. The defects can be classified into many categories depending on the process followed. It can also be grouped as minor, major and severe defects.
Step 4: The developer makes the changes to the design depending on the defects recorded during the review process.

Terminology and Assumptions:

In any formal work, it is very important to have precise definition and to explicitly state assumptions. In this paper also, an attempt has been made to keep these terminology and assumptions very simple.

The following definitions and terminology are taken from reference[1]. This will give the reader a better idea of the topic under discussion.

Defect or a **Bug** is an issue that is captured and logged while unit testing or system-testing phase.
Review [1]: This denotes a more or less formal examination of the specifications, code and other deliverables from a software project.
Review Cycle [1]: This term denotes the normal chain of approach, which the specification, plans and the other deliverables must go through.
Inspection [1]: This term denotes a rigorous technique by Michael Fagan of IBM Corporation. Inspection covers requirements, design, code, user documents and even test plan. Inspection has the highest efficiency of any known form of defect removal - greater than 75%.

For the reviewing team, the optimum range is greater than 60% - 75%. The company goal can have different bands to categorize the efficiency to suit its purpose. A study in US was published (Reference [2]) which depicted the defect removal efficiency in different stages of life cycle of the software is listed below.

Life Cycle Stage	Efficiency
Requirements	77%
Design	85%
Coding	95%
Testing	80%

In organizations like Microsoft, AT&T and Hewlett-Packard, efficiency at these stages is as high as 96%, and with their teams' best efforts, it is up to 99% (Source Reference [2]). If the development team, consisting of the developers and the reviewers, follows this review process then why do we still find design faults and design defects in the later stages? It is clear that the developers did not design the system as per the requirements. However by having an effective review team, it is possible to eliminate these design flaws during the review process. Thus, it is essential to find a method of evaluating the efficiency of the person or the team involved in reviewing. How effective the review was at the design-requirements phase? Did the review team adopt the right method and foresee the defects? By formulating a method, it will be helpful in distinguishing between an effective and a not-so-effective reviewer. Consistency of reviewer efficiency is also a major factor in the review process. In some large group review it may be

possible that few reviewers may not be effective, mainly because of the negligence factor. Hence it is also essential to see the consistency of the reviewer's efficiency with group size.

An attempt has been made in this paper to find such a method to evaluate the effectiveness of the reviewers in the development cycle review process by experimenting some simple models.

- Measure the effect of group "individuality" on review efficiency.
- Measure the individual's performance in review.
- Measure the variation of "competency level" of the team with efficiency.
- Measure the effect of group size on efficiency of offline review

Knowledge Experience Point (KEP) is a new term proposed in this paper, which refers to an index scale. The value of KEP for each reviewer is decided on the basis of years of experience in the organization and years of experience in the project, skill, domain knowledge and training. It is observed that with the increase in years of experience, after a time, the reviewing efficiency will not vary significantly [7]. This essentially means that the difference in the contributions by a person with experience of 4 years and that of 6 years is not as significant as the difference in the contributions by a person with 2 years experience and that of 6 months. The experience is classified into two different cases.

Calculation of KEP: A simple model is used to calculate the KEP of the reviewer. The model is based on the experience of the reviewer in software engineering and project experience, which includes parameters such as skills, domain experience etc. Three years experience in software industry is considered to be the threshold of maximum knowledge for understanding the process. As explained earlier, the experience gained beyond 3 years will not contribute to any significant change to the efficiency of the review [7]. However this assumption is based on a sample study. The org-wide trends could vary slightly. The value addition in terms of expected efficiency for different level of experience is shown in Fig 2. The performance of the reviewer cannot always be 100%, but can tend to 100% [2].

Most of the development projects are of the duration of 3 months to 24 months. The statistics taken from a sample of 23 software engineers (working experience from 1 year to 7 year) from 6 different teams is represented in Fig. 2. The data showed that efficiency increases drastically within 3 months to 12 months, and gradually sustains to a constant tending to 100%. It is interesting to note that the results were comparable to those obtained by Xenos and Christodoulakis [7].

Experience in Project	KEP	Efficiency
0 – 3 months	0	5 – 20
3 – 6 months	1	20 – 30
6 - 12 months		

Fig 2 Experience versus Efficiency in reviewing design documents

Table 1. KEP versus Project Experience

Experience in Industry KEP Efficiency	
0 – 3 months	0 < 20
0.5 yr – 1 yr	1 30 – 45
1 – 2 years	2 50 –70
3 and higher	4 80 -95

Table 2. KEP versus Software Eng. Experience

Experience KEP	Software Engineering Project
3 years 2 year 4	
3 years 2 months 3	
2 year 6 months 3	
6 months 3 months 1	
1 year 4 months 2	
2 year 1 month 2	
1 months 1 month 0	

Table 3. KEP from Software Engineering and Project

Experience

Based on these data, a simple KEP is proposed, and the values are shown in Table 1 and Table 2. The values are sample data. These can be modified to suit the project conditions. However to substantiate the analysis, a case study is considered.

It should be noted that the Project experience is a subset of industry experience. If a person has 1-month industry experience and 1-month project experience, it means that he has joined the project directly. Then, he is as good as a fresher with no prior experience. Thus his KEP is 0.

In tables 1&2, the maximum value of KEP is 4 and minimum value is 0.The accuracy of the end results tends to increase if the KEP is tuned to higher scaling. A 10-level KEP scaling gives better results than a 4-level KEP scaling. Different organizations may have different ways of arriving at a KEP, and its fidelity will depend on the sophistication of model used by the organization. Based on the notion of KEP, several experiments can be devised which measure various aspects of review effectiveness.

Suggested Experiment One:

Measuring the effect of group "individuality" on review efficiency.

The simplest measure of group size is the number of people in the group. If we have a group of 10 people during a review, then it would be difficult to say that the contribution of all the 10 individuals would be the same in terms of their efficiencies as reviewers. People with different backgrounds and experience participate in the review process. And again in the same group, it is easy to find that one or few reviewers are experts and the others are relative greenhorns. The interesting task now would be to evaluate how the review effectiveness of a group changes as we change its composition from, say 5 people consisting of one expert and 4 greenhorns, to a group of 5 reviewers who are "equals", but not quite as good as the expert. We then need a different measure of the "individuality" of the team. A possible measure is the quantity.

J (individuality) is defined as the ratio of maximum KEP of the individual participating in the group and the sum of the KEP index for the same group.

$$J = \frac{\text{Max } \{K_i, 1 \le i \le n)\}}{(K_1 + K_2 + \ldots + K_n)} \quad \text{---------- (1)}$$

Where n is the group size, and K_i is the i^{th} member's KEP. We may call this the "normalized maximum KEP". Higher values of J correspond to dependency of one or few reviewers.

Example: Module 1 has two reviewer (R1 & R3) whose KEP are 4 and 4 respectively. Module 2 has two reviewers (R1 & R2) whose KEP are 4 and 1 respectively.
Then J for Module 1 for reviewer 50% and for Module 2 is 80%

Consider two groups G1 and G2 doing the review for a module. Let both G1 and G2 have 5 reviewers each. From KEP table let us assume that the reviewer's index is computed for both the groups as follows.

Group G1	KEP
G1-R1	1
G1-R2	1
G1-R3	2
G1-R4	2
G1-R5	4

Group G2	KEP
G2-R1	1
G2-R2	3
G2-R3	4
G2-R4	2
G2-R5	4

Applying equation (1) in both the cases, we get the value of J for G1 and G2 as 0.4 and 0.2 respectively. Thus it is certain that the dependency of the effectiveness of the review is less in case of G1 than in case of G2.

If the KEP indices are measured and quantified by including all the parameters and assigned from 1 to 100, then probably this data would be very significant in measuring the impact of the expert or "individual" in the review. If we have a group review results database, then we can isolate a sample for the experiment by identifying a set of groups, which have approximately the group size, and similar or different KEP total. If we compute J for each of these groups in the sample, and plot the review efficiency versus J, then the results will indicate whether, for example, you can expect similar results as with an individual expert reviewer if you have a group of reviewers with the same expertise "scattered" or "distributed" amongst the members.

It should be noted that the idea here is not to highlight the significance of values of J with respect to the effectiveness of the team, but just to focus the impact of "individuality" effect on the review team. Using this method, one can determine the importance of an expert being in the group review. So if the PL is aware of the KEP, then using this data the review team size and grouping can be formed.

Suggested Experiment Two

Measuring the reviewer's efficiency as an individual in a group.

This section is essentially the most useful factor for the team leader. The common method of evaluating an individual's performance in review is to track his individual contribution. However in this section, emphasis is laid on tracking the individual's review effectiveness and efficiency with respect to the KEP index.

Review Hypothesis: The efficiency and effectiveness of a reviewer is consistent irrespective of group size. This above statement is very critical as we assume that the contribution of the reviewer is 100% for any review, which he is capable of, and with any group size.

If we have to measure the efficiency and consistency of a reviewer, R then following the above hypothesis, irrespective of the size of the group, the contribution, efficiency, effectiveness should not vary drastically. This is assuming that all the modules in which the reviewer R participates are of similar type. If we have the data where reviewer R_1 participates in reviewing modules M_1 to M_n, long with Reviewers R_2 to R_m, as shown in the table below. The values of KEP are represented in the table for each of the reviewers.

Table 4: KEP index for different reviewer against the module

	R_1	R_2	R_3	R_4					R_m
M_1	4								
M_2	4	1							
M_3	4	1	1						
M_4	4	1	1	2					
M_n	4	1	1	2					4

M -> Module; R is reviewer

Each module will have the contribution from each of the reviewer's KEP. Let us assume that Module M_1 is reviewed by only R_1, and some modules reviewed by many reviewers who have same and similar KEPs. By calculating the sum of KEPs for each of the module, and plotting it for Reviewer R_1, we get the following data, which will vary between 100% and to 0% (0% indicates no participation).

From (1) , the % Contribution of reviewer R_1 for Module M_n =

$$J(n) = \frac{\text{KEP of } R_1}{\text{Sum of KEPs of all reviewers for module } M_n}$$

(Where n varies from 1 to N.)

If DI (n) is the number of defects injected in the module M_n, where n is any value between 1 to N DD (n) is the number of defects detected in the module M_n during review, where n is any value between 1 to N. If DE (n) is the number of defects escaped in the module M_n, then the review efficiency of each module from our basic mathematics is

$$E_f(n) = \frac{DD(n)}{(\sum (DD(n) + DE(n)))*100} \quad \text{(Where n = 1 to N)} \qquad \text{---------- (2)}$$

Plotting the values of J and E_f for different values of n, the review efficiency of R_1 can be determined. The table below is an ideal example of reviewer R, who is efficient in all the reviews irrespective of the group size.

Modules	J	E_f
M1	100	100
M2	80	100
M3	60	100
M4	40	100
M5	20	100

Example A2: Module 1 has two reviewer (R1 & R3) whose KEP are 4 and 4 respectively. Module 2 has two reviewers (R1 & R2) whose KEP are 4 and 1 respectively. ***Then PK for Module 1 for reviewer R1 is 4/8 or 50% and for Module 2 is 4/5 or 80%***

What will happen if we do not have values of a reviewer's data as an independent, meaning when the reviewer R is doing it all alone?

In these cases, a ***suggested method*** *is to normalize the highest J, and measure the trend. Now normalize the values of J (n =1, m) to* ***NzJ*** *for different module for each resource. The normalization helps in finding a common index for comparing the efficiency of different reviewers. Normalized KEP of the reviewer also helps in identifying the performance of the reviewer in the group. From the above example A2, the reviewer 1's contribution in module 1 is 50% and in module 2 is 80%, and by normalizing against 80%, the study of the reviewer capability can be predicted at 100% contribution or when he is doing it alone.*

Note: It is important to go for normalization only when the absolute value of J is above 50%, else the whole purpose is defeated to identify the individual contribution.

Significance of Low Normalized J (NzJ)

If the review group or the team is large (the reviewers are of more than 4 or 5 for any module), then PK will have low value. Example: 4/(4*5) = 0.25, which reflects the performance of the individual in a large group having similar KEPs.

Significance of High Normalized J (NzJ)

When the value of normalized PK is high, the pattern of the review efficiency is related to the individual contribution of the reviewer or the reviewer's contribution is large in the small group. A typical example is having a senior person reviewing a new, inexperienced review team. Plotted NzJ signifies the contribution of the reviewer at different levels. So typically at 100%, the reviewer will be the only person reviewing.

Viewing the Trend in reviewing.
If a plot is drawn between J and Ef, then one can study the pattern of the reviewer's efficiency at different levels of contribution in the group. The consistency of the reviewer is also indicated in the graph. High turbulence and wavy graph shows inconsistence behavior of the reviewer. Again, if the reviewer efficiency is fluctuating between 100 to 80% for all NzJ, then the reviewer is very consistent.

Calculating the Review Efficiency
The idea of mapping the NzJ to the module efficiency is to study the characteristics of the reviewer's performance in a large or small group review. Thus it is essential to find an index or parameter for calculating the review efficiency. In ideal case, the performance of the reviewer does not depend on the size of the review group, or on any external factors. So for ideal reviewer, the efficiency of is a constant 100%. This does not change when the value of group size changes. In other words, the efficiency is a constant value with respect to NzJ.

If NzJ of the reviewer for different modules m varies from 100% to 0% (when the reviewer does not participate in the review, then the NzJ for that module is 0%), and the E_f (m) is the efficiency of the corresponding modules, then
Normalized Review Efficiency Index (REX) is defined as

$$REX = \frac{\sum (NzJ(m).\ E_f(m))}{\sum (NzJ(m))} \qquad (3)$$

If the module's efficiency is 100% at NzJ for values 100% to 0%, then the reviewer's efficiency is also 100%.

Individual and Group Reviewer Efficiency
In the plot, the efficiency of the modules is plotted against the NzJ, or the normalized contribution. As mentioned earlier, the higher values of NzJ reflects the performance of the reviewer as an individual, and lower range to the group review characteristics.
Using equation (3), it is possible to derive a new parameter to measure the reviewer's efficiency in a group and as an individual. Based on the inputs of the Project Lead / or the org-wide norms, if 75% NzJ is considered to be the threshold divide, then ***Individual REX (IREX)*** and ***Group REX (GREX) are*** calculated from (3)

Individual REX (IREX)	Group REX (GREX)
$IREX = \frac{\sum (NzJ(m).E_f(m))}{\sum (NzJ(m))}$ (Where 100 >= NzJ (m) > 75) (4)	$GREX = \frac{\sum (NzJ(m).E_f(m))}{\sum (NzJ(m))}$ (Where 75 > NzJ (m) >= 0) (5)

The results in the above equation would be same even if absolute values of J are considered, and corresponding REX, IREX and GREX signify the performance of the reviewer in absolute terms. However if value of J is lesser than 50%, then it is not recommended to take normalization. However, this again depends on the org-wide goals and standards.

Note: The value of NzJ=75 is a sample, and it can be changed as per the project requirements and org-wide quality goals.

What does IREX and GREX signify to the Reviewer's Consistency?

Today, the reviewer's consistency and performance are very important factors in successful project completion with minimum or zero defects. Whenever the group of the review team is big, the following things can happen.

- Few people are serious about the review process.
- Contribution made by the review team cannot isolate the performance of the individual, as it becomes very tedious to capture the data of every individual reviewer.
- Similarly, reviewer's who are serious when they review as an individual can behave differently when they are part of a big group. This is mainly because of the mental block that other reviewer's can find the defects.

Thus it is very important to study the behavior of the reviewer's in a small group and in a large group.

As per our hypothesis that performance of the reviewer's efficiency is consistent at all times, these should not occur in a review process.

The value of IREX and GREX can signify this consistency. If **IREX = REX = GREX**, then the reviewer is consistent for the size of the review group. If the value of IREX, REX and GREX are in the range 55-60%, still it is **valid to say the reviewer is consistent though not efficient**.

However if the value of IREX is 95% and that of GREX is very low, say lower than 75-70%, then it implies that the reviewer is more serious as a single or dominant reviewer, and not that serious when it comes to group. However this is a debatable topic, as to what level can the GREX be for the above statement to be valid. In a world-class quality company, the ideal reviewer should be consistent, and the difference between the IREX and GREX should not vary much. In some cases it is possible to find that IREX is of small value, and GREX to be high value. In those cases, the possible explanations are that the reviewer as an individual is not efficient, and because of some serious reviewer's the GREX is high.

When will the above IREX, GREX and results from the graph not be true?

If reviewer R, has a value of J for M1 as 100% and then for M2 is 20%, and the corresponding efficiency of the modules from equation (2) are 100% and 100%, and if R accepts the fact that she was serious and contributed fully during review of M1 and did half-heartedly during M2, but just because of X in M2 who was serious, the module was 100% efficient, then the above experiment III, analysis will not be true.

CASE STUDY - A

Data is collected from a development project having 10 developers and 12 at peak. The review performance is tested using the data. The Table 5 represents the KEP for each reviewer, and reviewer's participation for each module, and the corresponding efficiency of the module. The efficiency is calculated using equation (2) for different modules. The value is set to zero if the reviewer has not reviewed the corresponding module.

Table 5: KEP index for different reviewer against the module

M	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	Ef
1	0	0	4	0	4	4	1	0	3	3	4	4	100.00
2	2	0	4	0	0	0	0	0	0	0	0	0	100.00
3	2	4	4	0	4	4	0	4	0	3	4	4	96.15
4	2	0	4	0	4	4	1	0	0	3	0	0	77.50
5	0	0	0	0	0	0	1	0	0	0	4	0	100.00
6	2	0	4	0	0	0	1	0	0	3	4	0	100.00
7	2	0	4	3	0	4	0	0	0	3	4	0	92.86
8	2	0	0	0	4	4	0	0	0	3	4	0	85.71

M -> Module; R is reviewer

Performance trend for Reviewers R3 & R11

Using the Table 5 data, equation (1), the values of J and NzJ are computed for reviewers R1 and R10 for all the modules.

Table 6: Values of J and NzJ for reviewer R3 and R11 for different modules, with Total KEP

Module	Total KEP	J (R3)	J (R11)	NzJ (R3)	NzJ (R11)
M1	27	14.81	14.81	22.22	18.52
M2	6	66.67	0.00	100.00	0.00
M3	33	12.12	12.12	18.18	15.15
M4	18	22.22	0.00	33.33	0.00
M5	5	0.00	80.00	0.00	100.00
M6	14	28.57	28.57	42.86	35.71
M7	20	20.00	20.00	30.00	25.00
M8	17	0.00	23.53	0.00	29.41

Plotting the graphs in Figure 2 and Figure 2, for reviewer R3 and R11 from table 6,

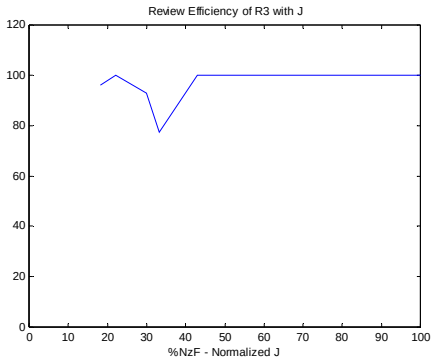


Fig 2: Normalized J (NzJ) plotted for Reviewer R3 versus Ef

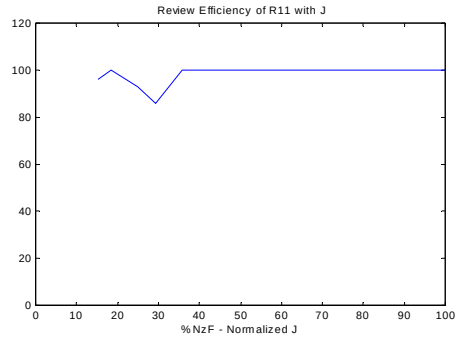


Fig 3:Normalized J (NzJ) plotted for Reviewer R11 versus Ef

Parameters	Reviewer R3	Reviewer R11
	Normalized J (NzJ)	Normalized J (NzJ)
REX	95.81	97.06
IREX	100	100
GREX	92.95	94.69

Analysis of Reviewer R3 and R11

The values of REX, IREX and GREX are tabulated above for reviewers R3 and R11. With these parameters, one can easily compare the performance of the reviewers R3 and R11. From the results, it is evident that both the reviewers have performed exceptionally well both as a individual reviewer as well as in a group. If we had the results for GREX lower than 90.00%, say for R3 [Example: GREX = 60%], then one of the possible conclusions is that R3's performance in the group is not very effective.

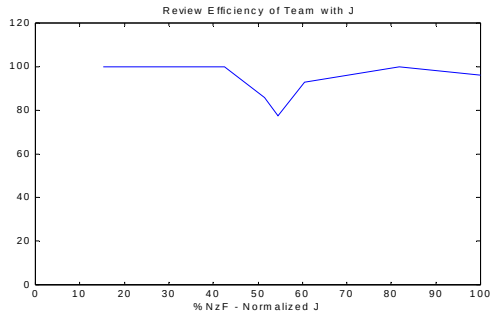
In the current project case, the performance of R3 and R11 are consistent irrespective of the group size. However if the data has to be really accurate, then R11 is a better performer than R3, since REX and GREX are higher for R11.

Suggested Experiment Three:

Measuring the review team's Efficiency with competency level

On the similar lines, by knowing the value of KEPs for different reviewers, it is possible to study the performance of the review team as a whole. Here the main idea is to compute the total KEP for each module, and measure against the efficiency using equation (1) and (2). Considering the case study A, table 6 represents the total KEP and the NzJ for different modules.

Module	Total KEP	NzJ (Team)	E_f
M1	27	81.82	100.00
M2	6	18.18	100.00
M3	33	100.00	96.15
M4	18	54.55	77.50
M5	5	15.15	100.00
M6	14	42.40	100.00
M7	20	60.61	



92.86

M8
17
51.52
85.71

Table 7: Ef for the team (4 level KEP)

Fig 4: Normalized J (NzJ) plotted for Reviewer R11 versus Ef

NzJ is computed by taking ratio of total KEP for each module and the maximum total KEP for any module. In our example, it is (Total KEP/33).

IREX for the team here means, the contribution made by the team when maximum reviewers (KEP is max, greater than 75% is an example) are involved, and for GREX it means lesser KEPs are involved.

The following is observed from the Table 7 and the figure 4, the efficiency of the Team on the whole from equation (3) is **REX = 93.44%,** and when more reviewers with high KEP are involved, then **IREX = 97.88%** and the **GREX** is measured to be **90.11%.**

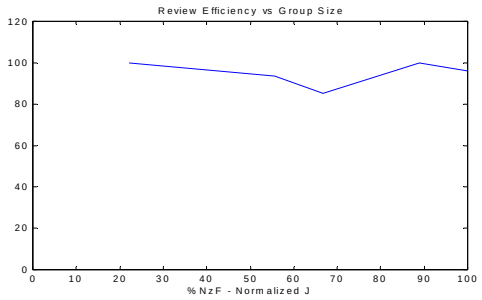
Suggested Experiment Four:

Effect of group size on efficiency of offline review

As mentioned earlier, for a group review, the first step is to send the document to all the reviewers for initial offline inspection. Now in this experiment, the focus is to study the reviewer's efficiency, in those group reviews where the size is large. If a reviewer knows that she is a part of a large group of reviewers with similar expertise, will she become more lax and hence less efficient during the offline inspection process? It is very human to have an attitude "in a group my negligence will not have an impact". Though as individual the reviewer may be excellent and efficient, it is quite possible that with the increase in the group size, the reviewer may become more lax. This is not always true, but it would be a very interesting study to find the same.

In this experiment, the focus is to study the "efficiency of group" with size in the review process. To study this, we may take a sample of group reviews involving "balanced" groups. Here, a balanced group means that all the reviewers in the group have more or less the same KEP index, and also the average KEP in the sample should be in the same range. While groups in this sample can be of different sizes, we ensure that the average KEP is the same across all groups. For example, a good sample of 3 groups would have the following individual KEPs:
Group 1: 4,5,6 (Average KEP- 5); Group 2: 5,5,5,6 (Average KEP - 5.25); Group 3: 4,4,5,5,6,6 (Average KEP - 5)

We would then plot the offline efficiency versus group size for each sample group. If there were a significant downward trend in efficiency with increase in-group size, this would indicate that the reviewers tend to be more lax when they know they are part of a larger group. Laitenberger et. al [9] proposed the hypothesis that the larger the effort for defect detection, the more defects are detected. Based on these results, a study was made for the finding the variation of efficiency with the size of review team, and not taking into the consideration of the experience. The Fig 7 is the plot for the Team size versus the review efficiency.



Team size versus Review efficiency

The value of **REX, IREX** and **GREX** was found to be **93.43%, 92.72%** and **94.89%.** The IREX for the NzJ (KEP) is higher than the IREX for the team size, which clearly signifies the fact that the review efficiency will improve if more experience people are involved, rather than increasing the team size. In other words, the increasing the NzJ value to the Team can get better results than increasing the Team size.

Summary and Conclusion:

In this paper, an attempt has been made to propose a new index known as knowledge experience point (KEP). The value of KEP is assigned to each reviewer by the PL on the basis of years of experience in the organization and years of experience in the project, skill, domain knowledge and training. The PL can decide the scaling of KEP to any degree based on the org-wide norms and standard. In the present study, the KEP was indexed between 0 - 4. Based on KEP, four experiments were suggested for devising different methods of analyzing the reviewer's characteristics and performance in a group and individual review. These models could be used for post project analysis of the performance of different reviewers with respect to group "individuality", individual, competency level and group size.

This method is applicable for all review process in Software lifecycle (design review, test plan review and test case review).

The review efficiency and the consistency of the reviewer are studied from the plots in the case study. This new KEP method can be a useful tool to predict the efficiency of the reviewers in the project, and can be used for evaluating the performance and provide a means for further improvement in the coming projects. The teams' reviewing efficiency and the pattern are also analyzed in this study. This method can be used for find the reviewing efficiency of the members of the client's and outsourcing teams, and to what level the expert reviewers of the client team have contributed to the review. Using the same experiment, one can also study the variation efficiency of the team as a whole with size and the competency level, rather than just the efficiency.

From the plots of efficiency and normalized KEP, some parameters were defined to extract the review efficiency index (REX), review efficiency index in a group (GREX) and as an individual (IREX). The level of consistency can be measured using the new derived parameters REX, IREX and GREX. The cut-off for IREX and GREX was taken as 60%, however the PL can decide this based on the org-wide norms. The IREX for the NzJ (KEP) is higher than the IREX for the team size, which clearly signifies the fact that the review efficiency will improve if more experience people are involved, rather than increasing the team size. In other words, increasing the NzJ value to the Team can get better results rather than increasing the Team size. These parameters can be used as a useful tool to find the consistency of the reviewer in a large group.

The accuracy of the project data (defects by modules, total defects, escaped defects, number of modules and reviewers, etc) is very critical for good estimation of the trends of reviewing efficiency. Further it is established that the accuracy of KEP will improve with higher scaling, probably 0-10. As an extension, the pattern study using Fourier analysis can give significant characteristics of the reviewer, which can be taken as a future work.

During the study it was observed that the trend and efficiency analysis using the KEP method, is not accurate for reviewers who participated in lesser reviews. The KEP method is more suitable for large projects with many modules, and reviewers participating in many modules. Also the KEP method may not be able to capture the real information in those conditions when the reviewer actual did not contribute in a group, yet the review efficiency was 100% because of effort from other reviewers. The current study is based on data of 3 projects, and the validity of this can be proved after studying more samples, which is in the process, and can be considered as a future work.

Acknowledgements
I want to thank the NCRSCOT Team for giving me the time to write this material, and NCRASUI and NCR-OSDC teams for providing valuable data. I also want to thank Ramkumar R, SetLabs, for inspiring me to think in a different dimension. Finally, I would like to thank my wife, Krithika for encouraging and inspiring me throughout the research.

References:

1. Caper Jones, "*Assessment & Control of Software Risks*", 1994 Prentice Hall Edition, pp. 603, 579.
2. Caper Jones, "*Pattern of Software System Failure and Success*", 1996 Thompson Computer Press, pp. 180-181.
3. Software Productivity Consortium, *Software Measurement Guidebook*, Chapter 6, 1995, Thompson Computer Press, pp. 116-119
4. Bhashyam M R, Meera G R, "*Group Review Procedure*", Oct 1999, Infosys Technologies Limited, QSD/REF/939
5. J.C.Kelly, J.S. Sherif, and J. Hops, "*An Analysis of defects densities found during software inspection*", J. Systems Software, Feb 1992, pp. 111-117
6. Schneidewind, N.F, "*An Integrated Process and Product Model"* Proc. of the Fifth International Software Metrics Symposium, November 20-21, 1998, Bethesda, Maryland, pp. 224-234
7. M. Xenos and D. Christodoulakis, "*Software Quality: The user's point of view*" Proceedings of First IFIP/SQI International Conference on Software Quality and Productivity (ICSQP 94), pp 266-272.
8. A. Porter, L. Votta Jr et al., "*Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment*", IEEE Transaction of Software Eng., June 1995, pp. 563-575.
9. Laitenberger, O.; Leszak, M.; Stoll, D.; Emam, K. E.: *Quantitative Modeling of Software Reviews in an Industrial Setting*. Proc. of the METRICS'99, Boca Raton, Florida, November 1999, pp. 312-322

About the Author

KiranKumar Marri has earned a B.E in Electronics and Communication from the Sri Venkateswara College of Engineering, University of Madras, and a M.S by Research in Biomedical Engineering from Indian Institute of Technology, Madras. He has Software experience in the fields of Biomedical Applications, Data Warehousing applications, Retail Market solutions, Self-checkout terminal solutions and Secure Transactions solutions over a span of 7 years. He is currently working as a Project Manager at Infosys Technologies Limited, Bangalore and leads a QA team in Product Competency Center.