White Paper: A Defect Prioritization Method Based on the Risk Priority Number

Authors:

Julie Cohen
Robert Ferguson
William Hayes

Problem to be solved

Most software systems have some "defects" that are identified by users. Some of these are truly defects in that the requirements were not properly implemented; some are caused by changes made to other systems; still others are requests for enhancement – improvements that would improve the users' experience. These "defects" are generally stored in a database and are worked off in a series of incrementally delivered updates. For most systems, it is not financially feasible to fix all of the concerns in the near term, and indeed some issues may never be addressed. The government program office has an obligation to choose wisely among a set of competing defects to be implemented, especially in a financially constrained environment.

Selecting the best set of fixes for the next incremental release is a difficult chore as there are many possible ways to prioritize the work. The release may be selected from defects that focus on the workflow of select system users. Since there may be distinct user groups, this method may generate conflict when a high-priority deficiency report (DR) is of great importance to one user community and fails to address concerns of other user communities. Alternatively, an increment (or part of an increment) may address the full set of issues related to a specific function of the system (e.g. mission planning). It is also possible to plan an increment to eliminate requirements for contractor field support, or some other existing "work-around" that is part of a complex workflow. A mathematical view of all the possible methods of selection quickly reveals there are hundreds of thousands (if not millions) of potential solutions to grouping, ordering and packaging releases.

This white paper provides a description of a generalized technique that could be used with any type of system to assist the program office in addressing and resolving the conflicting views and creating a better value system for defining releases. This technique was developed with the help of a Department of Defense program office.

Method being adapted

The Failure Modes and Effects Analysis (FMEA) method employs a measurement technique called Risk Priority Number (RPN) to quantify the relative priority for addressing known failure sources. The

technique relies on quantifying three distinct attributes of failure sources in a composite measure which helps to structure objective analysis and decision making.

> **Severity**: a rating of the adverse impact of the failure – a measure that reflects the negative consequence to the system users due to the "defect."
>
> **Occurrence**: how often the source of failure is encountered – a measure that reflects one element of cost for the persistence of the failure.
>
> **Detection**: how detectable the failure is when it occurs – a measure that reflects the risk of unmitigated consequences if the failure is not remedied.

These three attributes are combined to form a composite measure of risk remaining in the system, which can be reduced by fixing defects. This measure of risk allows decision makers to assess the value returned from doing the work. Because RPN is correlated to the cost of failure and correction, a sum of RPN values for all defects can be used as a measure of product quality.

Development of an RPN methodology for Deficiency Report (DR) prioritization

This section describes the steps we used with one specific customer, but there could be other ways to approach this effort. We began by forming a working group made up of personnel from the Program Office, Federally Funded Research & Development Centers (FFRDCs), a University Affiliated Research Center (UARC), Configuration Control Board members, and user command representatives. The group did not include end users due to workload issues, but it might have been more effective had end users been involved.

We started with the three broad areas of severity, occurrence, and detection. Working with customer subject matter experts to fine tune these areas enabled us to better reflect the nature of the system, and the types of defects under consideration. In the area of severity, after many discussions, we ended up with four main attributes to assess: Data Fault Condition, System Crash Condition, System Function Condition, and Operational Impact. In the area of detection we used User Visibility, Data Issues, and Security Risk. For Occurrence we looked at how often the issue occurred coupled with the time to either recover or to work around.

The next step was to develop rating scales for each area. We used a combination of 6-point and 5-point scales. The scale used for Operational Impact was: 6 Certain mission failure, 5 Could cause mission failure, 4 Certain delay/limit to mission operations, 3 Could delay/limit mission operations, 2 Increases operator workload significantly, and 1 Increases operator workload slightly. Similar rating scales were formulated for each area. It is important to point out that specific definitions (which are most likely specific to the software system) are associated with each point in the scale – people are not simply asked to "choose a number from 1 to 6."

After we had rating scales for each area we needed to develop proportional scaling factors that would assure that the impact of the 1-6 rating was appropriate for each level. For the 6-level rating we used factors of 1, 1.5, 2, 4, 8, and 24. For the 5-level rating we used 1, 2, 4, 8, and 24. These ratings were

considered in light of the definitions of each point on the scale. The subject matter experts were asked to consider, for example, "is the second category really twice as severe as the first, and the third one twice as bad as that?" For the occurrence rating we multiplied the number of occurrences per year by the larger of the recovery or work around time. These were then given a scaling factor as shown in Table 1 – Time Scaling Factors.

Description	Occurrence Time Range	Scaling factor
More than a week	2400+	24
A full week	961-2400	8
Up to two days	481-960	4
Up to a full day	61-480	2
Up to an hour	16-60	1.5
Brief Interrupt	0-15	1

Table 1 – Time Scaling Factors

In the FMEA formulation, RPN is a product of the three categories. Our working group chose to use a weighted sum so that specific weightings could be more transparent. This could have been accomplished by using different scaling factors, but we felt it would be easier to explain the method to the user community if we used the weighted sum method. We went through several iterations of the weightings. After testing the weightings with dummy data and samples of actual DR data, we established the following procedure.

In the area of severity, we took the largest value (or most severe condition) for Data Fault Condition, System Crash Condition, and System Function Condition and weighted this by 10%. We left Operational Impact as its own category since it has great importance to the users and we weighted this 30%. This results in the severity category having a 40% overall weight. In the area of detection, we took the largest value for User Visibility, Data Issues, and Security Risk and weighted that by 30%. The Occurrence area was also weighted 30%. This resulted in a range of RPN from 0 to 2400. Note that this is not a continuous scale from a statistical perspective.

We used sample data to test the scales and exercise the analysis process with the working group. This step allowed exploration of the operational usage of RPN values along with other data such as

- cost to repair the DR (SLOC is generally used for cost in this paper)
- system function or element the DR belongs to
- expressed priority of the DR from the user groups (for example B-2 would be user B's second rated priority)
- association of DRs with other attributes such as skills required to fix and test the defect (which may be more diverse in complex military systems)

The goal is to get user data from real DRs to compute RPN and perform analyses to assist the Program Office and the using command in prioritizing DRs to be corrected in the upcoming increments. A sample of the analysis that can be performed is described in the next section. While DR prioritization is the main reason for implementing RPN in the program we were working with, there are also some secondary benefits. The RPN data can help users to better communicate the issues associated with the

defects to the program office and the contractor. In addition, it provides a uniform method of assessing DRs across all users and all functionality.

Example of DR Analysis

In looking at the analysis possibilities we developed four basic methods, as shown in Table 2 - Analysis Methods.

Method	Brief Description	Pros	Cons
Functionality	Group DRs by system function using RPN and Source Lines of Code (SLOC) to select order	- Easier to test specific functional areas - Should see improvements in specific areas addressed	- May not address top user ranked DRs - Some functional areas will not be addressed in every increment - Some functional areas may still need to be split due to SLOC constraints
System Risk	List DRs by RPN and draw a line at the amount of SLOC per release; Best used for pure maintenance (regression testing only)	- Addresses system level risk first - Fairly easy to use	- Doesn't specifically address functionality groups - Doesn't specifically address user rankings
User rankings	List DRs by user rankings and draw a line at the amount of SLOC per release	- Addresses user rankings - Fairly easy to use	- May fix DRs with lower overall system risk earlier; Doesn't address system value - Doesn't specifically address functionality groups - Need to address differences between users
Hybrid	Combinations of the methods above	Depends on method	Depends on method

Table 2 - Analysis Methods

For this white paper we have created a set of non-customer data, just to illustrate the methodology. We created 26 DRs. These DRs are for four separate user groups (A-D), with RPNs ranging from 220 – 2400 and Source Lines of Code (SLOC) ranging from 20 – 1500. In addition, the DRs cover three different functional areas (admin, set-up, and planning) with three different "need dates" (Initial Operational Capability (IOC), Full Operational Capability (FOC) and Post FOC. The contract calls for software releases every 6 months that include up to 2500 Source Lines of Code (SLOC).

We will illustrate the hybrid method in this white paper. We suggest starting with a simple graph of RPN vs. SLOC to get a general idea of the distribution of the DRs. This is shown in Figure 1.

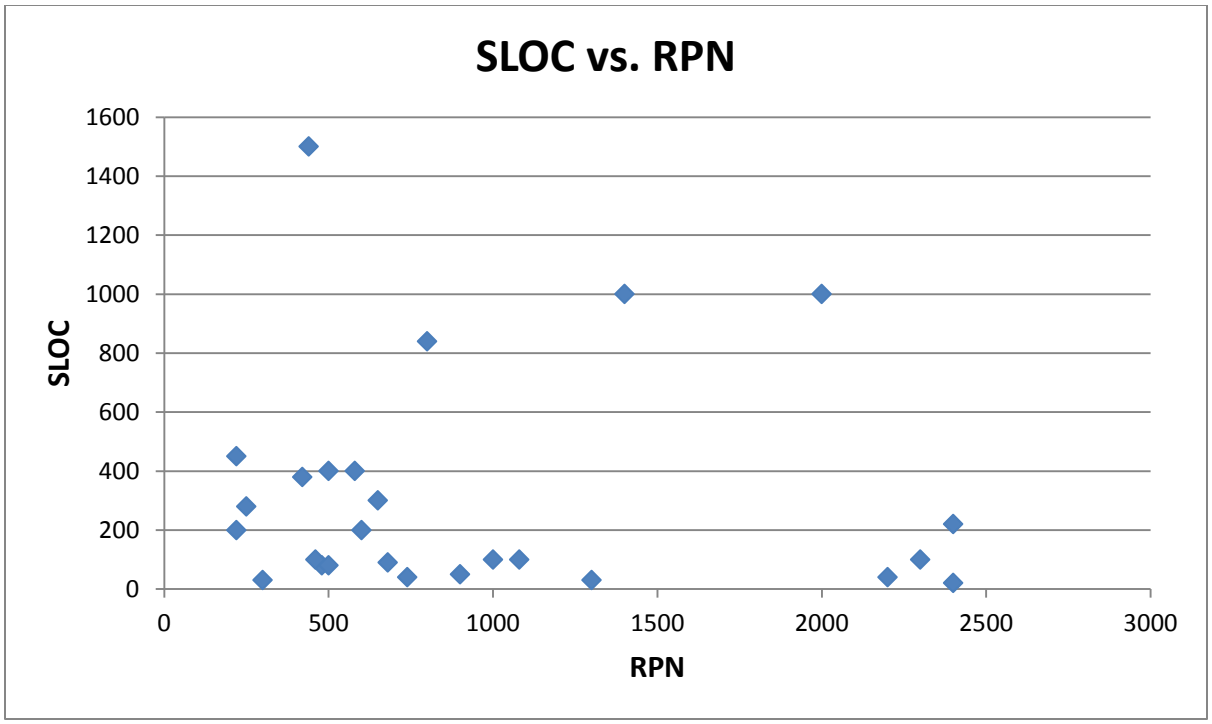

Figure 1 - SLOC vs. RPN

You can see that there are a few High RPN/Low SLOC DRs in the bottom right, several high SLOC DRs scattered across the RPN range, and a range of DRs with lower SLOC at the bottom the figure.

Another general view is shown in Figure 2 - SLOC vs. RPN by User. This is the same graph as Figure 1, except it uses separate symbols for each user. You can see that there is very little grouping (in terms of RPN or SLOC) across the users. That means, there is no user group where all the DRs have notably consistent RPN or SLOC values – all users have DRs that cover a range of each measure.

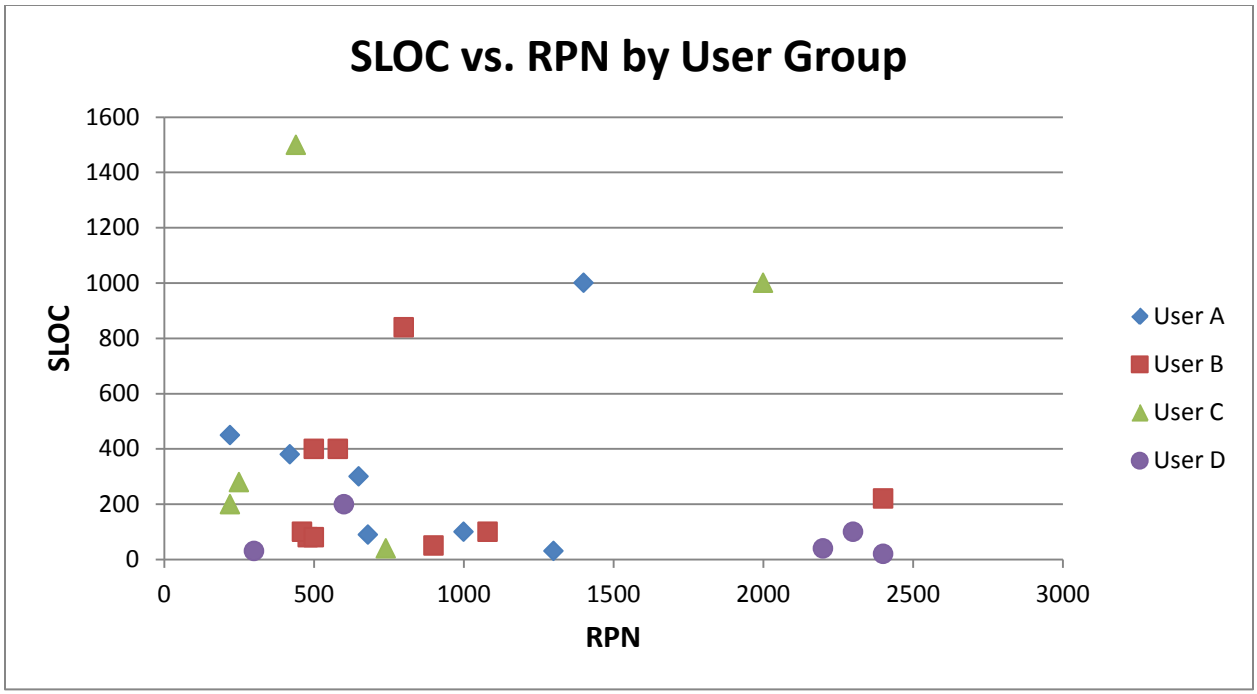

Figure 2 - SLOC vs. RPN by User

For this analysis we will assume we can include 2500 SLOC in each increment and that we have two increments prior to IOC, one additional increment prior to FOC and up to two increments post FOC. We will start the hybrid analysis with the most important factor at the current time. What constitutes the most important factor will likely change over time, so the analysis would need to be re-accomplished at various points in the program. We will assume that at this point in the program, the most important factor is the need date for the DRs that are required prior to IOC.

The DRs that are needed prior to IOC are shown in Table 3.

DR #	User	User Priority	RPN	SLOC	Need Date	Function
102	A	A-3	1000	100	IOC	Planning
103	B	B-1	800	840	IOC	Security
104	C	C-4	250	280	IOC	Planning
109	C	C-3	440	1500	IOC	Planning
124	B	B-8	500	80	IOC	Set-up
TOTAL SLOC				**2800**		

Table 3- DRs Needed Prior to IOC

This makes it clear that two increments will be needed to complete the IOC DRs. If we look at the RPN numbers for these DRs, we would want to try to fix the highest RPN DRs in the first increment (shown in Table 4) to reduce the most risk as soon as possible.

DR #	User	User Priority	RPN	SLOC	Need Date	Function
102	A	A-3	1000	100	IOC	Planning
103	B	B-1	800	840	IOC	Security
124	B	B-8	500	80	IOC	Set-up
109	C	C-3	440	1500	IOC	Planning
104	C	C-4	250	280	IOC	Planning
TOTAL SLOC				**2800**		

Table 4 - DRS needed prior to IOC by RPN

Since implementing the top four DRs would put us over the 2500 SLOC limit at 2520 SLOC, we would include DRs 102, 103, 124 and 104 to get under the 2500 SLOC limit. This adds up to 1300 SLOC and would reduce a large amount of system level risk first. We would now look to the FOC DRs to complete increment 1. These are shown in Table 5.

DR #	User	User Priority	RPN	SLOC	Need Date	Function
SLOC from DRs 102,103, 124, 104				1300		
108	D	D-4	2400	20	FOC	Admin
112	D	D-2	2200	40	FOC	Set-up
107	A	A-2	1400	1000	FOC	Set-up
113	A	A-6	650	300	FOC	Admin
117	A	A-5	420	380	FOC	Admin
Total SLOC				3040		

Table 5 - DRs needed prior to FOC by RPN

When we add the first 3 FOC DRs highlighted in yellow (108, 112, 107) to the 1300 SLOC already accounted for we get 2360 SLOC. Which would leave some margin to reduce risk, or we could continue to look for additional DRs to get to the 2500 SLOC limit. In this example we chose not to do that. Our first increment (shown in Table 6 - Final DRs in Increment 1) would include 7 DRs with a total RPN of 8550 out of a total RPN for all DRs of 24,820. So we have addressed almost 34% of the overall system risk in the first increment.

DR #	User	User Priority	RPN	SLOC	Need Date	Function
102	A	A-3	1000	100	IOC	Planning
103	B	B-1	800	840	IOC	Security
124	B	B-8	500	80	IOC	Set-up
104	C	C-4	250	280	IOC	Planning
108	D	D-4	2400	20	FOC	Admin
112	D	D-2	2200	40	FOC	Set-up
107	A	A-2	1400	1000	FOC	Set-up
Totals			8550	2360		

Table 6 - Final DRs in Increment 1

For the second increment we would start with the remaining IOC and FOC DRs shown in Table 7.

DR #	User	User Priority	RPN	SLOC	Need Date	Function
109	C	C-3	440	1500	IOC	Planning
113	A	A-6	650	300	FOC	Admin
117	A	A-5	420	380	FOC	Admin
TOTAL SLOC				**2180**		

Table 7 - Remaining IOC and FOC DRs

We can still add more DRs to this increment, as the SLOC does not yet total 2500. If we look at the remaining DRs shown in Table 8, we may want to look at something other than RPN to select the next DRs for this increment. Since there are only two "Admin" DRs remaining, if we assume it is important to the users to get specific functionality capabilities implemented, we would add DR 100 and 120 (highlighted in yellow) to the second increment and this would increase the SLOC from 2180 to 2250. Since we still have SLOC available, we might add the highest RPN DR that will fit and that is DR 101 with 220 SLOC (highlighted in green).

DR #	User	User Priority	RPN	SLOC	Need Date	Function
101	B	B-2	2400	220	Post FOC	Planning
115	D	D-3	2300	100	Post FOC	Planning
106	C	C-5	2000	1000	Post FOC	Planning
100	A	A-1	1300	30	Post FOC	Admin
125	B	B-9	1080	100	Post FOC	Planning
119	B	B-3	900	50	Post FOC	Security
120	C	C-1	740	40	Post FOC	Admin
121	A	A-7	680	90	Post FOC	Security
111	D	D-1	600	200	Post FOC	Set-up
116	B	B-6	580	400	Post FOC	Security
105	B	B-4	500	400	Post FOC	Set-up
114	B	B-7	480	80	Post FOC	Planning
122	B	B-5	460	100	Post FOC	Security
118	D	D-5	300	30	Post FOC	Planning
110	A	A-5	220	450	Post FOC	Planning
123	C	C-2	220	200	Post FOC	Planning
TOTAL SLOC				**3490**		

Table 8 - Remaining Post FOC DRs

The DRs implemented in Increment 2 are shown in Table 9 - DRs Implemented in Increment 2 and in the first two increments we have reduced the total system RPN by 58%.

DR #	User	User Priority	RPN	SLOC	Need Date	Function
109	C	C-3	440	1500	IOC	Planning
113	A	A-6	650	300	FOC	Admin
117	A	A-5	420	380	FOC	Admin
120	C	C-1	740	40	Post FOC	Admin
100	A	A-1	1300	30	Post FOC	Admin
101	B	B-2	2400	220	Post FOC	Planning
Totals			5950	2470		

Table 9 - DRs Implemented in Increment 2

Since completing specific functionality is important, we have sorted the remaining DRs by function (shown in Table 10 - Remaining Post FOC DRs). We can see that the planning area has the highest overall RPN count, and can fit into the 2500 SLOC limit. We could also add the first three DRs from the Security area, or, if user priority is also important we could consider adding DR 111 (D-1) under the Set-up Area to address User D's highest priority DR. That would mean working off less system level risk, but if the user priority is also important at this point, that may be a good trade-off.

DR #	User	User Priority	RPN	SLOC	Need Date	Function
115	D	D-3	2300	100	Post FOC	Planning
106	C	C-5	2000	1000	Post FOC	Planning
125	B	B-9	1080	100	Post FOC	Planning
114	B	B-7	480	80	Post FOC	Planning
118	D	D-5	300	30	Post FOC	Planning
110	A	A-5	220	450	Post FOC	Planning
123	C	C-2	220	200	Post FOC	Planning
			6600	1960		
119	B	B-3	900	50	Post FOC	Security
121	A	A-7	680	90	Post FOC	Security
116	B	B-6	580	400	Post FOC	Security
122	B	B-5	460	100	Post FOC	Security
			2620	640		
111	D	D-1	600	200	Post FOC	Set-up
105	B	B-4	500	400	Post FOC	Set-up
Totals			1100	600		

Table 10 - Remaining Post FOC DRs

Regardless of which DRs are added to complete Increment 3, the remaining DRs can all be worked off in the last increment prior to FOC.

Conclusion

There are many ways to use RPN in conjunction with other system priorities when trying to work off system defects. This white paper gives one simple example. We hope that this one example will allow the readers to extrapolate to their own system and think of ways that RPN would assist in assessing overall system risk and help to prioritize DR work-off.

Another major use for RPN is during continuing sustainment after FOC. The sum of the RPNs for the collected DRs is a means to assess changes in the overall quality of the product. Used this way, the Sum of RPNs is a better measure of change in product quality than the defect density since it is possible to improve defect density by reducing only the number of very easy fixes. It is also a better means to improve quality than by ordering DRs by user priority because

- Order by user priority alone may introduce and exacerbate conflict between the various users and with other stakeholders
- User priority is not necessarily correlated to broader measures of functionality or performance
- A reduction in the total number of defects (alone) is not likely to be sufficient to instill confidence by users or sponsors

Since Sum(RPNs) is tied to elements of cost of error (quality) a significant reduction in Sum(RPNs) between releases should satisfy stakeholders that quality is improving. Certainly, if the Sum(RPNs) is increasing with new releases, it will become cause for alarm.

Since the Sum(RPN) is a measure of overall quality (value), but not the only view of quality when applied in the hybrid analysis, we can judge that the multiple viewpoint method considers a measurable definition of quality and is adjusted by multiple views of functionality and performance while satisfying the necessary cost constraints. When the program office and the user community priorities are exposed via the hybrid method, the negotiations of priority are likely to be open and objective. In this situation it is more difficult to use political influence to override decisions outside the main discourse.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0000760