

New Risk Scoring Model

The “Risk” IssueType is new here at [COMPANY]. It should be used when there are issues relating to bugs, enhancements, or general factors in a release (such as time constraints or integration dependencies) that could have impact on the operational risk of the targeted system. This IssueType, plus the new Linking feature within QC 9.2, will provide a great deal of value to IT and GROUP Managers in pre-release go-no-go decision-making, and will become integral to our Change Management process (as sign-off on application-specific production readiness reports become required before CRs can be approved for production deployment). Below is a detailed breakdown of the scoring, calculation model (simple as it is) and how the resulting “Impact Score” has been baselined and scaled.

Severity Key	Score
1 - System Failure	4
2 - Function Failure - no Workaround	3
3 - Function Failure - with Workaround	2
4 - Moderate to Minor Issue	1

Priority Key	Score
1 - Urgent	4
2 - High	3
3 - Medium	2
4 - Low	1

Probability Key	Score
Percent Chance Exposure will occur	01-100

Exposure Key	Score
10. Broad System Failure	10
9. Broad System Impact/Issues	9
8. External System Failure	8
7. External System Impact	7
6. Local System Failure	6
5. Local System Impact	5
4. External System Issues	4
3. Local System Issues	3
2. Minimal System Impact	2
1. Minimal System Issues	1

As you can see above, we use some “native” fields (**Severity, Priority**) to represent some scoring factors. In other issue tracking systems that capture more “native” data such as inter-system dependency, mean time to recovery and high availability system status, those fields are calculated against a user input value for probability that the issue will emerge to generate an “opportunity cost” and from that a scaled value (to be used as a KPI, etc). Since QC doesn’t have that type of data natively, we’re requiring that users put in a pair of values for Risk IssueTypes to help generate the raw score (**Probability, Exposure**). As with Severity, we’ve done our best to set an objective scale that matches the representative value. The raw formula is as follows:

$(\text{Severity Score}) * (\text{Priority Score}) * (\text{Probability}) * (\text{Exposure}) = \text{Opportunity Cost}$

From there the **Opportunity Cost** is scaled to represent an **Impact Score**. Several benchmarks were used to calibrate the thresholds for various Impact Scores, and from that a KPI “Grade” of HIGH, MODERATE, and LOW was established.

Below is a series of benchmarking scenarios we used to calibrate the Impact Scores.

The first benchmark - Severity 1, Priority 1, 50% Probability, with 5 - Local System Impact: $4 * 4 * 50 * 5 = 4000$

This **Opportunity Cost** value was mapped to an **Impact Score** of **5**, and graded (along with all Impact Scores above it) with a HIGH rating – ensuring that any high severity/priority issue that had more than a coin-flip of a chance to severely impact production would receive the proper scrutiny (and would likely hold up the release if the IT or GROUP manager opted not to sign off on a production readiness report with a Risk item with a “HIGH” Graded Impact Score.)

The second benchmark – Severity 1, Priority 1, 75% Probability, with 10 - Broad System Failure: $4 * 4 * 75 * 10 = 12000$

This **Opportunity Cost** value was mapped to an **Impact Score** of **10**.

This is more to set a general “high reasonableness” score against the various factors we are multiplying out. The tendency would be to see issues with lower severity would receive a lower impact score as well – and vice versa. That’s why the top 40%ish of the scale spans two values. The Probability score can finesse the raw score, and the Exposure value can move it in “chunks”.

To set a reasonable median value between the two, we calculate the value for a scenario that falls reasonably between the two impact scores:

Severity 1, Priority 1, 50% Probability, with 9 - Broad System Impact/Issues: $4 * 4 * 50 * 9 = 7200$

The effect of this type of issue tracking (particularly as it relates to configuration, deployment, operation/continuity and security issues) usually has the effect of spawning quite a bit of negotiation amongst the team to account for each factor, including the “native” factors (Severity and Priority) that are often spawned from Defects or Enhancements. If there’s a Defect that’s a Severity of 3 and a related Risk item with a Severity of 1, then there’s some basic match to work out. Likewise, it’s a highly valuable gauge as issues are raised and mitigation strategies are brought to bare. And by contrast, increases to probability and exposure can be increased as the release sign-off approaches and long-standing issues go unresolved. This is a very useful replacement to the common practice of using a “ring and run” email where there’s no tracking, no formal sign-off, and often the most important decision-makers are left out of the exchange.

Opp Cost	Impact Score	Grade
0	0	
500	1	LOW
1000	2	LOW
2000	3	MODERATE
3000	4	MODERATE
4000	5	HIGH
5000	6	HIGH
6000	7	HIGH
7200	8	HIGH
10000	9	HIGH
12000	10	HIGH
16000	MAX	HIGH

Data, Reporting, and KPIs

QC Dashboard was a major disappointment for us. It’s small wonder that HP has EOL’d their Dashboard in lieu of completely re-architecting the solution natively to QC v10 (in their future roadmap). Since we planned to leverage SharePoint 2007 for publishing production readiness reports and other static artifacts, we decided to “double down” our bet and integrate QC directly with MS Report Services to publish data directly to SharePoint as a Dashboard. There are several templates that help streamline setup, but once the primitives are in place, there will be some additional configuration in order to populate data through the various databases

Single Sign-on

As part of our migration to a new platform, it was decided to integrate single sign-on functionality so that any valid QC user could log in under their system-managed credentials. Once successfully logged in to Quality Center, the user is