Consortium for IT Software Quality

Sample Acceptance Criteria with CISQ Standardized Metrics

The Consortium for IT Software Quality, a neutral and non for profit organization founded by the Software Engineering Institute (SEI) at Carnegie Mellon University and the Object Management Group (OMG), has standardized a set of rules for measuring the non-functional characteristics of software. CISQ defines these rules for Security, Reliability, Performance Efficiency, and Maintainability, with coding (unit-level) and architectural (system-level) attributes. CISQ has also standardized an automatable Function Point measure for sizing.

The CISQ Quality Characteristic rules are consistent with ISO/IEC 25010 definition. They are designed to be automated on source code to identify critical vulnerabilities in the software that are severe enough that they need to be fixed. Combined with a sizing measure, a density metric is produced for each quality characteristic. Thresholds can be set for each characteristic.

The CISQ Quality Characteristic Measures cover eighty-six well-established software engineering rules to ensure secure, reliable, efficient and easy to maintain software. The table at http://it-cisq.org/standards/ shows a snapshot of software engineering rules contained in the measurement of each quality characteristic at the unit level and system level.

1. Acceptance Process for Software Deliverables

a) Supplier shall forward all completed deliverables (source code files, database scripts, configurations and all other source-level components needed to compile the deliverables into an executable system) to the CLIENT Program Manager or designated signatory.

b) Once the Deliverable is presented for Product Readiness Review (PRR) to CLIENT, CLIENT shall have ten (10) business days from that time to either:

   i) accept the Deliverable in writing (or)

   j) reject the Deliverable by notifying Supplier in writing of CLIENT's reasons why the Deliverable is not acceptable.

c) If the deliverable is rejected or returned along with rejection/return of the document, CLIENT shall identify specific non-compliance issues or areas to be corrected in order to enable Supplier to accomplish corrections.

d) In the event CLIENT rejects a Deliverable, Supplier will address CLIENT's reasons for such rejection and resubmit the Deliverable to CLIENT for acceptance at no additional charge.

Comment [t1]: This can be done at Test Readiness Review (TRR) as well

Comment [t2]: Ensure appropriate time period

e) Notwithstanding the foregoing, a Deliverable shall be deemed accepted upon the occurrence of CLIENT failing to accept or reject the Deliverable within ten (10) business days of receipt.

Comment [t3]: Ensure appropriate time period

f) CLIENT may accept a deliverable which does not comply with CISQ rules for any reason, such as valid technical justification from the supplier or arbitrage between time to market and structural quality.

g) If, after a commercially reasonable number of attempts to modify the non-conforming Deliverable by Supplier, CLIENT still, by written notice to Supplier, rejects the Deliverable, CLIENT may terminate this SOW pursuant to Section 22.1 (Termination for Cause) of the Agreement, effective upon notice to Supplier. Upon termination as set forth in this section and subject to payment by CLIENT of all fees for Services rendered hereunder through the effective date of termination, CLIENT and Supplier shall have no further obligation to Supplier pursuant to this SOW.

Comment [t4]: Refer back to the Termination Clause in the main contract

h) CLIENT will evaluate the deliverable quality

   a. Deliverables will be automatically rejected if the delivered code violates any of the Critical Rules as defined by CISQ specifications.

   b. Code quality will be measured against the following measures:

Application Quality Measure	**Measurement Criteria**
Reliability	Delivered codebase will incorporate error handling as per the approved design and adhere to the 29 rules in the CISQ Reliability specification.
Security	Delivered codebase will not introduce any new security vulnerability issues and as specified by the 22 rules in the CISQ Security specification.
Performance Efficiency	Delivered codebase will not introduce any new performance efficiency issues, as specified by the 15 rules in the CISQ Performance Efficiency specification.
Maintainability	Delivered codebase will have maintainability criteria (*i.e.*, flexibility to make changes easily) implemented as per the 20 rules in the CISQ Maintainability specification.

   c. CLIENT may use any necessary manual inspections and CISQ-conformant technology to perform such reviews.

   d. CLIENT and Supplier may agree in advance to use a specific subset of the CISQ rules that are particularly relevant to the CLIENT's objectives.

The following section can be used as either a performance incentive to drive further improvement in quality of software assets, or as an "at-risk" calculation, permitting the CLIENT to receive some deliverables that are not completely compliant with CISQ rules, but to charge a penalty to the Supplier upon doing so. Typically the recommended "at-risk" amount is the same as the recommended performance incentive amount – 10% of contract value.

Comment [t5]: Note to contracting personnel.

Appendix: Performance Incentives or "At-Risk" Contract Amount

Should the Supplier meet the expectations of deliverable quality for the given release, that is meet the target scores for specific Application Quality Measures as defined below, then CLIENT agrees to release to Supplier an incentive payment. Performance incentives will be calculated based on the analysis done on accepted code for production, based on the CISQ specifications.

Analysis is to be performed both at the System Level and the Unit Level. System level means that interactions between application components are considered in determining the compliance to CISQ rules. The target metrics will look at the number of CISQ rule violations per amount of software in the application. It will also look at newly introduced violations. The goal will be to improve the violation density, while ensuring that no “New Violations” are introduced in each release. This analysis will employ a static analysis tool. Criteria for incentive payment are outlined in the following table:

Maximum incentive: 10% of the billing for the given release

Application Quality Measure	Analysis Level	Weights for Incentive	Expected Target *
Reliability	System	25%	0.1
Reliability	Unit	5%	0.1
Security	System	30%	0.02
Security	Unit	5%	0.02
Performance Efficiency	System	20%	1
Performance Efficiency	Unit	5%	1
Maintainability	System	5%	3
Maintainability	Unit	5%	3

* Expected Target is calculated based on the agreed upon improvement over previous baseline. This target is calculated as a violation density, that is the number of CISQ violations per Automated Function Point (AFP). If AFPs are not being calculated, then KLOC (thousand lines of code) can be substituted in the denominator.

Performance Incentive review process will be done as follows:

1) Monthly Performance Reviews
   a) After each release performance reports are generated
   b) CLIENT delivery managers meet with Supplier to review performance
   c) Review any Critical Deliverables due or missed
   d) Perform RCA (root cause analysis) for missed Application Quality Measures
2) Quarterly Changes
   a) CLIENT meets to set priorities and agree on changes
   b) Weighting of individual service levels
   c) Promote or demote Critical Rules
   d) Add or delete Application Quality Measures
3) Resets
   a) Expected targets can be reset based on performance during past release or year
   b) Expected Performance Improvement: New Expected Targets = Current Expected Target + 10% of the delta between the then current Expected and Perfection

Example of incentive payment:

Billable amount for the release: $1,000,000.00
Maximum incentive amount: 10% = 10% X $1,000,000 = $100,000

Application Quality Measure	Weightage for Incentive	Potential Incentive ($)	Expected Target	Target for New Violations	Actual Score	New Violations	Incentive Payment
Reliability – System Level	25%	25%*$100,000 = $25,000	0.1	0	0.09	0	$25,000
Reliability – Unit Level	5%	5%*$100,000 = $5,000	0.1	0	1.2	0	0
Security – System Level	30%	30%*$100,000 = $30,000	0.02	0	0.018	5	0
Security –	5%	5%*$100,000 =	0.02	0	0.011	0	$5,000

Comment [t6]: 10% is for illustration. Actual incentive amount needs to be defined

Comment [t7]: The weights shown here are for illustration. These should be changed to meet your needs.

Comment [t8]: Expected Target is calculated based on the agreed upon improvement over previous baseline. It is recommended to calculate baseline score based on 3-5 previous releases. In the absence of 3-5 releases, data from the most recent scan can be used.

Comment [t9]: Data shown here are for illustration purposes. Actual baseline data should come from the actual application analysis.

Comment [t10]: RCA can be performed if the supplier is consistently not able to meet specific measures.

Comment [t11]: This is the recommended approach, but can be changed based on project needs.

Unit Level		$5,000					
Performance Efficiency – System Level	20%	20%*$100,000 = $20,000	1	0	0.89	0	$20,000
Performance Efficiency – Unit Level	5%	5%*$100,000 = $5,000	1	0	2.64	0	0
Maintainability – System Level	5%	5%*$100,000 = $5,000	3	0	3.59	15	0
Maintainability – Unit Level	5%	5%*$100,000 = $5,000	3	0	2.73	0	$5,000
Total	100%	**$100,000**					**$55,000**

Example of Reset and Performance Improvements:

Application Quality Measure	**Current Expected Target**	**New Expected Target** = Current Expected Target + 10% of the delta between the then current Expected and Perfection
Reliability – System Level	0.1	0.1 – 10% *0.1 = 0.09
Reliability – Unit Level	0.1	0.1 – 10% *0.1 = 0.09
Security – System Level	0.02	0.02 – 10% * 0.02 = 0.018
Security – Unit Level	0.02	0.02 – 10% * 0.02 = 0.018
Performance Efficiency – System Level	1	1 – 10% *1 = 0.9
Performance Efficiency – Unit Level	1	1 – 10% *1 = 0.9
Maintainability – System Level	3	3 – 10% *3 = 2.7
Maintainability – Unit Level	3	3 – 10% *3 = 2.7

Comment [t12]: Perfect score would be a violation density of zero