

Definitions for ICT testing metrics

Department of Health Standard

QH-IMP-439:2020

1. Statement

Capturing quality metrics from ICT testing activities is essential to inform decision making and test process improvement.

At an Enterprise level it is especially important to adopt standard definitions for key ICT testing metrics so that reporting across diverse systems and projects remains consistent, allowing meaningful comparisons to be made.

2. Scope

This standard applies to all employees, contractors and consultants within the Department of Health divisions and business units.

This standard can be used by Hospital and Health Services either as is, by re-branding or as a basis for a Hospital and Health Service's specific standard

3. Requirements

When test metrics are recorded from testing activities, the data collected should be able to be translated to the following definitions for Enterprise reportability. Metrics captured may be *named* differently to the standard metrics but should have the same *meaning* so that the data is translatable to the standard metrics.

This standard does not require that *all* metrics defined in this document are captured.

3.1. Test environment logical definitions

Test environment metrics (about how current environments are being used) can be used to better inform Enterprise strategy and planning for future improvements.

| Test Environment Category | Definition |
|---------------------------|--|
| DEV | <p>The DEV environment contains functions that may provide systems support to inform or assist analysis and design activities, such as prototyping and demonstration exercises.</p> <p>These functions enable assessment, education and options analysis prior to the actual commitment to build new capability or change existing functionality.</p> <p>This environment provides an important opportunity to iterate through analysis and design, incorporating customer review and feedback. This can be fundamental to minimising issues during requirements gathering and design, as well as understanding and setting customer expectations.</p> <p>Common levels/types of testing: Unit, Static review.</p> |

| Test Environment Category | Definition |
|---------------------------|---|
| BUILD | <p>The BUILD environment is the starting point for development and/or configuration of change, rollout and/or enhancement packages that are destined for implementation into the production environment.</p> <p>The environment contains functions that support these development activities, including Unit testing. This is an important function within the BUILD environment that serves as a quality assurance check for all changes, signalling readiness to progress to the TEST environment.</p> <p>Common levels/types of testing: Unit, Static review, Build.</p> |
| TEST | <p>The TEST environment contains all functions required to perform formal functional and non-functional testing and is used to confirm tested features and functions are ready to progress to the Transition or Production environments.</p> <p>The purpose of this formal testing is to ensure that the changes meet their business/design goals, are safe to use, and will not harm the production environment.</p> <p>The environment is typically separated from production environments on production-like infrastructure (e.g. separate network, storage, firewall etc.).</p> <p>Used for most types of testing (including potentially destructive testing if sufficiently separated from production environments).</p> <p>Common levels/types of testing: Build, Cyber security, Data migration, End to end integration, Exploratory session, High availability, Network sociability, Performance, Regression, Sociability, Static review, System, System recoverability, User acceptance, Witness test execution.</p> |
| TRANSITION | <p>The TRANSITION* environment contains all functions required to support approved packages as part of their implementation into the production system, including staging, validation, and deployment.</p> <p>(*TRANSITION is generally known as the Pre-production Acceptance Transition environment (PAT))</p> <p>The environment is typically on production infrastructure (e.g. the same Network, Storage, firewall, using same H/W, O/S etc.) and with production-like data.</p> <p>Used primarily to rehearse production implementations. Not used for potentially destructive testing impacting production. All testing under strict production service management control.</p> <p>Common levels/types of testing: Cyber security, High availability, Implementation dress rehearsal, Static review.</p> |
| TRAIN | <p>The TRAIN environment contains all functions required to provide a complete training capability. This includes current production training and new release training.</p> <p>Please note that this environment does not provide a demonstration capability (this is provided in the DEV environment).</p> <p>May be used for testing purposes. Operational (production) databases containing security classified information or any personal / private information are not to be used in a TRAIN environment.</p> <p>Common levels/types of testing: None</p> |

| Test Environment Category | Definition |
|---------------------------|--|
| PROD | <p>The PROD environment contains all functions required to operate the production environment:</p> <ul style="list-style-type: none"> • Ensure system accuracy • Maintain and support users • Maintain the technical environment • Provide high availability and/or disaster recovery • Analyse and manage issues and/or problems. <p>Used for production validation testing that cannot be completed in other test environments (e.g. limited production deployment, or pre-go-live testing). Not used for potentially destructive testing impacting production. All testing under strict production service management control.</p> <p>Common levels/types of testing: Production validation.</p> |

3.2. Test level definitions

Test level* metrics (about the levels/types of testing occurring in the organisation) can be used to better inform Enterprise strategy and planning for future improvements. (*Also known as 'Test Products'.)

| Test Level Category | Definition |
|------------------------|---|
| Build | Build testing is testing performed by the group responsible for creating a repeatable process to build or configure a system or component to confirm the process is repeatable. |
| Cyber security | Cyber security testing is testing conducted in accordance with the Cyber Security Group Information Security Policy Framework. Cyber security testing occurs in relation to a system, device or data to mitigate or identify cyber security risks. |
| Data migration | Data migration testing is the confirmation that all migrated data has been accurately migrated to the target environment without impacting expected functionality. |
| End to end integration | End to end integration testing is testing performed on the system that specifically targets end to end integration (i.e. data exchange) with external/other systems to build confidence that integration business processes can be completed using the solution. |
| Exploratory session | Exploratory session testing is an approach to testing whereby testers are involved in minimum planning and maximum test execution in a controlled time-boxed testing session. Exploratory session testing is most useful when there are unclear requirements and/or when time is severely limited. The planning for exploratory testing requires the creation of an 'exploratory test session charter' (a brief declaration of the scope of a short time-boxed test activity, the objectives, and possible approaches to be used). Exploratory session testing is intended to complement NOT to replace other formal testing and relies on expertise, experience and intuition of the tester. |

| Test Level Category | Definition |
|--------------------------------|---|
| High availability | High availability testing is testing performed on the system to build confidence that the solution's infrastructure high availability features will work as expected in the event of failure of a redundant system component. |
| Implementation dress rehearsal | An implementation dress rehearsal is testing of the ICT change implementation plan steps (including reversion steps) performed by the group(s) responsible for implementing a system or system change into production. |
| Network sociability | The purpose of the Network sociability test (also known as wan worthiness testing) is to comprehensively analyse new systems to the QH network and make technical recommendations in order to mitigate potential impacts to existing network systems. |
| Performance | Performance testing is testing performed on the system to evaluate the degree to which it accomplishes its designated functions within constraints of time and resources. Examples are stress testing, volume testing, soak testing, endurance testing, peak load testing, and scalability testing. |
| Production validation | Production validation testing (PVT) is testing conducted after a system has been deployed to a production (or pre-production) environment, before widespread use of the system. |
| Regression | Regression testing is testing following changes to features or functions of a system (or to its operational environment) to identify whether unintended impacts have occurred to features or functions not intended to be impacted by the changes. |
| Sociability | Sociability testing is intended to ensure that a system will perform in its intended shared environment without adversely impacting (or being adversely impacted by) unrelated existing ICT systems. |
| Static review | Static review testing is verification of the documented work products produced to deliver a system. (These work products are typically produced before the system itself has been delivered and therefore detection of defects early in these work products can be much more cost effective). Examples are code review, peer reviews, and formal / informal reviews. Examples of work products reviewed are requirements and design documents, internal and vendor test deliverables, test cases, security and documents. |
| System | System testing is testing performed on the delivered system to build confidence that the solution is fit for purpose for the relevant Queensland Health business. |
| System recoverability | System recoverability testing is testing performed in relation to the system recovery plan of a system to build confidence that the system can be recovered quickly in the event of system failure. |
| Unit | Unit testing is software verification and/or validation of software code to gain confidence that individual units of source code are fit for purpose. A unit is the smallest testable part of a system. |
| User acceptance | User acceptance testing is testing conducted by or on behalf of (by delegated authority) end business users of the system to build confidence that the end business users' complete business processes can be completed. |

| Test Level Category | Definition |
|------------------------|---|
| Witness test execution | Witness test execution testing is a visual inspection by QH representatives of the execution of testing by another testing party (the 'witnessed testing party') in order to build confidence that QH requirements have been adequately tested. |

3.3. Test execution status definitions

Test execution status metrics are captured to inform up-to-date status reporting of test execution activities.

| Test execution status | Definition |
|-----------------------|---|
| Passed | The test has been executed and the expected result achieved. |
| Failed | The test has been executed and the expected result was not achieved. A defect should be raised in this instance. |
| Not run | The test has not yet been executed. |
| Blocked | The test cannot be executed because of a preventing (blocking) circumstance outside of the control of the tester. The blocking circumstance should be stated as a comment next to the test. |
| Not completed | The test execution was commenced but not completed. The reason should be stated as a comment next to the test. |
| N/A | The test is not applicable for the current test cycle and should not have been scheduled. The reason should be stated as a comment next to the test. |

3.4. Defect severity definitions

Defect severity metrics are captured to inform the assessment of success criteria for testing, and to help measure the value of testing efforts.

| Defect severity | Definition (Business Impact) | Definition (Technical Impact) |
|-----------------|---|---|
| 1 – Critical | The failure causes a system crash or unrecoverable data loss or causes impairment of critical system functions. The customer cannot continue using the system. No acceptable work-around exists. Business impact is Major or Extreme. | The failure prevents deployment of the system into production or deployment of the system severely impacts stability of the production environment. |
| 2 – High | The failure causes impairment of system function. The customer can still use the system but cannot perform a critical task and no acceptable workaround solution exists for that task. Business impact is Moderate. | The failure impacts implementation of the system into production or deployment of the system impacts production stability of system and/or other systems. |

| | | |
|-----------------|---|---|
| 3 – Medium | The failure causes impairment of system function or component. The customer can use the system (an acceptable workaround exists) but the defect is very annoying. Business impact is Minor. | The failure causes localised impact on implementation of the system into production or localised impact on production stability of the system and/or other systems. |
| 4 – Low | The failure causes inconvenience or cosmetic in nature. Long-term work-around for use in production is acceptable to the customer. Business impact is Negligible. | The failure causes no impact on implementation of the system into production and no impact on production stability of the system and/or other systems. |
| 5 – Observation | The issue raised is not classified as a failure of the system or component. The issue may be a point of clarification or potential future change request | Not related to a fault condition. No impact on implementation of the system into production and no impact on production stability of the system and/or other systems. |

3.5. Defect status definitions

Defect status metrics are captured to inform up-to-date status reporting of defects.

| Defect status | Definition |
|--------------------|--|
| New | The initial state for a defect that has been logged. Indicates that the defect has not been triaged or assessed. |
| Open | The defect has been assessed and confirmed (i.e. is valid). The defect is targeted to be resolved within the current release. |
| Rejected | The defect has been assessed and rejected by the defect triage process because it has been determined not to be a defect (i.e. is invalid). A defect root cause classification is captured to inform test process improvement. |
| Deferred | The defect has been assessed by the defect triage process and it has been determined that the defect does not require resolution in the current release but may require resolution in the future. If required, an appropriate workaround or action plan will be documented and communicated to the change and release implementation process. The defect will be transferred to the product backlog process to be considered for resolution sometime in the future lifecycle of the product. |
| Request for Change | The defect has been assessed by the defect triage process and it has been determined to be a request for change. The request will be transferred to the product backlog process to be considered for inclusion sometime in the future lifecycle of the product. |
| Ready for test | The defect has been resolved by the group or individual responsible and is now available for testing by the testing group. |
| Verified Fixed | Retesting of the defect shows the defect is no longer present. |
| Failed | Retesting of the defect shows the defect is still present. |

| Defect status | Definition |
|---------------|--|
| Reopen | A defect was incorrectly closed or has failed retesting and needs to be reassigned to the group or individual responsible for resolution. Reason for reopening must be stated in the defect log. |
| Closed | The defect requires no further action, now or in the future. The test manager (or delegate) has validated that the defect closure process has been completed successfully. A defect root cause classification is captured to inform test process improvement. |
| On Watch | The defect has been assessed and temporarily placed 'On Watch' because the defect is unable to be reproduced and is no longer evident in the test environment, even though a specific fix for the defect was not applied. The defect may be intermittent, may have been fixed as a result of another fix applied, or may have been reported incorrectly. The 'On watch' status is applied to monitor the defect for potential reoccurrence during a defined monitoring period. |

3.6. Defect root cause definitions

Defect root cause metrics are captured to inform process improvement. (Defect validity is included for clarity and is not a required metric.)

| Defect validity | Defect root cause | Definition |
|--------------------|---------------------|--|
| Confirmed defect | Code error | An error in the way the system code has been developed resulting in the system not behaving in its required way. |
| | Configuration error | An error introduced during configuration of the system. |
| | Current PROD issue | An error that already exists in PROD, is an acknowledged issue, and has been logged in the Production issue/problem management process. |
| | Hardware error | An error that was introduced because a hardware component of the system has failed. |
| | Specification error | An error that was introduced because a specification (e.g. a requirement or design document) was ambiguous, incomplete or otherwise failed to preserve the intent of the business requirement. |
| Unconfirmed defect | Unable to Replicate | An error that was reported but could not be replicated. The root cause is not known. |

| Defect validity | Defect root cause | Definition |
|-----------------|---------------------------------------|---|
| Not a defect | Current PROD functionality | Observed behaviour that already exists in PROD, is not considered to be an issue, and no Request for Change is required. |
| | Duplicate | A defect that had already been logged and is being managed under the original defect record. |
| | N/A for QH Workflow | Observed behaviour that may appear to be incorrect but does not impact QH workflow, is not considered to be an issue, and no Request for Change is required. |
| | New requirement not previously stated | Expected behaviour (that the defect reporter wished or expected) that is not in the current build because it had not been formally requested. A new Request for Change may be required. |
| | Test case error | An error that occurred due to an issue with the design of the test. (The test case was incorrect.) |
| | Test data error | An error that occurred due to incorrect test data – such as test data supplied with the test case, or test data stored in the system. (The test data was incorrect.) |
| | Test environment issue | An error that occurred due to the test environment not being in a valid state for the test. (The test environment was incorrect.) |
| | Tester or user error | An error that occurred due to an issue with the execution of the test by the tester or user. (The test case, data, and environment were correct but there was an error in execution.) |
| | Rejected | Observed behaviour that has been triaged and confirmed not to be a defect – no other root cause was applicable. |

| Defect validity | Defect root cause | Definition |
|-----------------|------------------------|---|
| | Unexpected Enhancement | Observed behaviour that is a new feature/enhancement of the system under test, is not considered to be an issue, and no Request for Change is required. |

4. Legislation

- *Financial Accountability Act 2009*
- *Hospital and Health Boards Act 2011*
- *Information Privacy Act 2009*
- *Public Records Act 2002*
- *Public Service Act 2008*

5. Supporting documents

- *Department of Health ICT Testing Policy QH-POL-439:2017*
- *eHealth Queensland ICT Testing Standard*
- *Data and application custodianship Policy QH-POL-469:2019*
- *Data and application custodianship: Roles and responsibilities*
- *Health Software – Part 1: General requirements for product safety (IEC 82304-1:2016)*
- *ICT Service Continuity Management Policy QH-POL-457:2018*
- *ICT Service Continuity Management Standard QH-IMP-457:2018*
- *Information Security Policy QH-POL-468:2019*
- *Information Security Standard QH-IMP-066:2:20155*
- *Medical Devices - Application of risk management to medical devices (ISO 14971)*
- *Medical device software – Software life cycle processes (IEC 62304)*
- *Risk Assessment and Treatment Guideline (included Risk Analysis Matrix) (QH-GDL-070-1-1)*
- *Software Testing (ISO/IEC/IEEE 29119)*
- *Standard for Software Reviews and Audits (IEEE 1028)*

Version Control

| Version | Date | Comments |
|---------|-----------------|--------------|
| 1.0 | 14 January 2020 | New Standard |