ISO/IEC/IEEE 29119-3 information items	Requirement (shall), recommendation (should), permission (may)
7.2.7.3 Test types	Shall
7.2.7.4 Test deliverables	Shall
7.2.7.5 Test design techniques	Shall
7.2.7.6 Entry and exit criteria	Shall
7.2.7.7 Test completion criteria	Shall
7.2.7.8 Degree of independence	Shall
7.2.7.9 Metrics to be collected	Shall
7.2.7.10 Test data requirements	Shall
7.2.7.11 Test environment requirements	Shall
7.2.7.12 Retesting	Shall
7.2.7.13 Regression testing	Shall
7.2.7.14 Suspension and resumption criteria	Shall
7.2.7.15 Deviations from the organizational test practices	Should
7.2.8 Testing activities and estimates	Shall
7.2.9 Staffing	Shall
7.2.9.2 Roles and responsibilities	Shall
7.2.9.3 Hiring needs	Should
7.2.9.4 Training needs	Should
7.2.10 Schedule	Shall
7.3 Test status report	Shall
7.3.2 Test status	Shall
7.3.3 Reporting period	Shall
7.3.4 Progress against test plan	Shall
7.3.5 Factors blocking progress	Shall
7.3.6 Test measures	Shall
7.3.7 New and changed risks	Shall
7.3.8 Planned testing	Shall
7.4 Test completion report	Shall
7.4.2 Summary of testing performed	Shall
7.4.3 Deviations from planned testing	Shall
7.4.4 Test completion evaluation	Shall
7.4.5 Factors that blocked progress	Shall
7.4.6 Test measures	Shall
7.4.7 Residual risks	Shall
7.4.8 Test deliverables	Shall
7.4.9 Reusable test assets	Should
7.4.10 Lessons learned	Shall
8.2 Test model specification	Shall
8.2.2 Unique identifier	Shall
8.2.3 Objective	Shall
8.2.4 Priority	Shall
8.2.5 Test strategy extract	Shall
8.2.6 Test model	Shall
8.2.7 Traceability	Shall

ISO/IEC/IEEE 29119-3 information items	Requirement (shall), recommendation (should), permission (may)
8.3 Test case specification	Shall
8.3.2 Test coverage items	Shall
8.3.2.2 Unique identifier	Shall
8.3.2.3 Description	Shall
8.3.2.4 Priority	Shall
8.3.2.5 Traceability	Shall
8.3.3 Test cases	Shall
8.3.3.2 Unique identifier	Shall
8.3.3.3 Objective	Shall
8.3.3.4 Priority	Shall
8.3.3.5 Traceability	Shall
8.3.3.6 Preconditions	Shall
8.3.3.7 Inputs	Shall
8.3.3.8 Expected results	Shall
8.4 Test procedure specification	Shall
8.4.2 Unique identifier	Shall
8.4.3 Objective	Shall
8.4.4 Priority	Shall
8.4.5 Start up	Shall
8.4.6 Ordered test cases	Shall
8.4.7 Relationship to other procedures	Shall
8.4.8 Stop and wrap up	Shall
8.5 Test data requirements	Shall
8.5.2 Unique identifier	Shall
8.5.3 Description	Shall
8.5.4 Responsibility	Shall
8.5.5 Period needed	Shall
8.5.6 Resetting needs	Shall
8.5.7 Archiving or disposal	Shall
8.6 Test environment requirements	Shall
8.6.2 Unique identifier	Shall
8.6.3 Test environment item	Shall
8.6.4 Description	Shall
8.6.5 Responsibility	Shall
8.6.6 Period needed	Shall
8.7 Test data readiness report	Shall
8.7.2 Unique identifier	Shall
8.7.3 Description of status	Shall
8.8 Test environment readiness report	Shall
8.8.2 Unique identifier	Shall
8.8.3 Description of status	Shall
8.9 Actual results and test result	Shall
8.9.2 Actual results	Shall
8.9.3 Test result	Shall

ISO/IEC/IEEE 29119-3 information items	Requirement (shall), recommendation (should), permission (may)
8.10 Test execution log	Shall
8.10.2 Unique identifier	Shall
8.10.3 Date/time	Shall
8.10.4 Description	Shall
8.10.5 Impact	Shall
8.11 Incident report	Shall
8.11.2 Timing information	Shall
8.11.3 Originator	Shall
8.11.4 Context	Shall
8.11.5 Description of the incident	Shall
8.11.6 Originator's assessment of severity	Shall
8.11.7 Originator's assessment of priority	Shall
8.11.8 Risk	Shall
8.11.9 Status of the incident	Shall

Annex B
(informative)

Overview of examples

B.1 Overview

Annexes C to R contain examples of the application of each information item (template) for agile and traditional projects, to demonstrate the applicability of this standard to both types of projects. These are examples only, and many variations are possible and likely.

The example test documentation is based on two fictional organizations:

— **Agile Corporation** is a large publication organization producing magazines and books. Agile Corporation has an internal IT department, which is responsible for developing and supporting the organization's IT products. Products are built by a single agile team and all development is conducted under an agile lifecycle. The organization has several years of experience working in this way and finds that it works well with their needs for new and enhanced IT systems to support their business. The organization has chosen to adopt an international standard for their testing, to demonstrate to their customers that they care about quality.

  The product featured in this example is the development of a new web-based magazine subscription system that allows customers to become subscribers and allows existing subscribers to change their personal information and order new or extended subscriptions.

— **Traditional Ltd** is a small company that produces advanced analysis equipment for the farming industry. Some of their products are safety-critical, in the sense that incorrect analysis results risk causing the prescription of incorrect fertilizer doses (either too much or too little). The organization is hence required to produce the product according to an international standard that demonstrates and retains evidence of correct behaviour during testing, along with solid test planning, traceability and reporting.

  The example project features the development of the PC-based portion of a product called UV/TRT-14 33a. It is an apparatus to measure fertilizer components and their concentration in earth samples. The apparatus has a user interface working on a PC with wireless connection to the measuring system.

The traditional examples are detailed, as they are designed to achieve full conformance (see 4.1.2). The agile examples are lightweight (i.e. minimal and lean), as they are designed to achieve tailored conformance (see 4.1.3), with some information communicated verbally. Agile Corporation's adoption is an acceptable form of tailoring, due to the lower perceived development, testing, quality and contractual risks involved with their product. The more 'heavyweight' documentation prepared by Traditional Ltd relates to a much higher risk system that requires far greater quality assurance and testing across the lifecycle. Any organization can adopt the level of test documentation that suits their teams and systems, from full conformance (i.e. production of all documents and inclusion of all fields) to tailored conformance with a minimal set of test documentation, with agreement from relevant stakeholders.

Not all examples in these annexes include all of the common information (see Clause 5), as the examples focus specifically on the testing-related contents of each information item.

The examples might not be internally consistent; each section is to be regarded as an independent example of the information related to the topic (heading).

The examples are not always complete. Where paragraphs or omitted text have been left out, this is marked by ellipsis, like this "…".

The word “shall” appears in some of the examples. These “shall”s are example wordings only and are not normative.

The examples do not follow the editorial rules in the ISO/IEC Directives, Part 2. Explanations on the examples are provided in NOTEs.

Annex C
(informative)

Test policy

C.1 Example 1 – Agile Corporation

Test Policy – Agile Corporation

Objectives of testing: To provide enough information to determine the quality and release readiness of each product under test.

Test process: The test process is based on ISO/IEC/IEEE 29119-2 and is aligned with the agile development methodology. All agile delivery teams are required to utilise automated Test-Driven Development (TDD) for unit testing and Acceptance Test Driven Development (ATTD) for system and regression testing.

Test organization: Testers are allocated to agile development (delivery) teams from a central pool, with each tester reporting to their agile Delivery Lead. Central experts (working under the Head of Testing) are also available to provide test consultancy services to each product as required.

Training: All testers are required to have an appropriate university degree or a minimum foundation-level industry certification in software testing. Testers are expected to be knowledgeable in agile concepts or to become so within three months of joining an agile team.

Ethics: All testers will adhere to the Tester's Code of Ethics

Standards: Agile Corporation utilises tailored versions of ISO/IEC/IEEE 29119-2 and ISO/IEC/IEEE 29119-3, with all tailoring decisions design to suit the organization's agile practices and all requiring approval by relevant the Head of Testing, Delivery Leads and Product Owners.

Related relevant policies: Software Development Policy for Agile Corporation, version 4.3 (12/2/2019).

Test process improvement & value determination: Retrospectives at the end of each iteration will capture lessons learned and improvement concepts and will reflect on the value of testing practices. Improvement actions will be actioned during future iterations. Innovations are shared with other agile teams for testing maturity uplift across the organisation.

Test asset archiving & reuse: all session sheets, documented tests & test automation frameworks will be retained according to the Release Test Plan for each team.

C.2 Example 2 – Traditional Ltd

Test Policy – Traditional Ltd

Objectives of Testing: Testing is conducted as a means for improving user and customer confidence in our products and determining when systems are ready for release. Testing is one of many means to achieve this goal.

Test Process: All testing is conducted according to ISO/IEC/IEEE 29119-2. The test process includes all test activities in the standard, namely: organizational test practices, test planning, analysis and design of test cases and procedures, specification of test environment and test data requirements, test execution and recording of the test including registration of any incidents, and test completion

and reporting. Static testing is also conducted on all projects according to ISO/IEC 20246. All software development projects must include a test project, with both projects starting at the same time.

Test Organization Structure: Each project will be allocated with testers that have specialist capabilities as required from a central pool. Each tester will report to their project manager and also independently to the Head of Testing as required (e.g. for escalation of testing-related risks or issues).

Tester Training: All testers are expected to have achieved a minimum university-level degree in computer science, computer technology, software engineering or similar course. All testers must also complete testing certifications that match their staffing level, which includes foundation level training for all staff, advanced test design for test analyst / senior test analysts and test management training for test leads and above.

Tester Ethics: All testers will follow the organizational code of ethics.

Standards: The organization follows ISO/IEC/IEEE 29119-2, ISO/IEC/IEEE 29119-3, ISO/IEC/IEEE 29119-4 and ISO/IEC 20246 (for Static Testing). The organization also follows the ISO/IEC/IEEE 12207 product development lifecycle.

Other Relevant Policies: All software development is conducted according to the Traditional Ltd Software Development Life Cycle, which is based on the above standards and can be found on the company intranet.

Measuring the Value of Testing: All projects are required to collect metrics that can be used to calculate the return-on-investment (ROI) for testing practices and new innovations. This can include calculating ROI in terms of reduced testing cost and effort and reduced time-to-market.

Test Asset Archiving & Reuse: All projects will store and maintain manual regression test suites in the centralised test management tool. All automated test scripts will be stored in the central code/version control tool. All manual and automated test scripts will be handed over to the business-as-usual (BAU) team at the completion of each project, after the warranty period is complete (using the company's standard knowledge retention and transfer process during handover).

Test Process Improvement: After each iterative release of a system to production, members of each project team, including all testers, will conduct a review of the testing and quality assurance practices conducted throughout the release, to identify improvements that can be implemented throughout the next release cycle. An annual test practice assessment will also be conducted, to determine whether existing testing practices are suitable and to identify new improvements that can be implemented within each project and as broader organizational testing improvement initiatives.

Issued by Miguel Sanchez (Head of Testing) and Ursula Myers (Head of Development)

Approved by Sarah Blacksmith (Chief Technology Officer)

Annex D
(informative)

Organizational test practices

D.1 Example 1 – Agile Corporation

Organizational Test Practices – Agile Corporation

Scope: The Organizational Test Practices are applicable to all testing conducted at Agile Corporation.

Risk management: All agile teams must conduct a product & project risk assessment based on the Testing-Related Risk Management Process (TRM56) defined on the team wiki. Risk assessments are conducted during iteration zero and revisited during subsequent iteration planning meetings.

Test selection & prioritization: All test selection and prioritization will be based on the iteration risk assessment, which will be carried out during each iteration planning session.

Test automation, tools, defect/incident management & test documentation: Tests are automated whenever possible via Agile Corporation's central automation framework, with scripts stored in the central source control system. Exploratory test sessions are captured via the approved recording tool, with execution evidence stored in the central test management tool. Manual tests (when required) are also stored in the central test management tool. All manual and automated tests must be traced to user stories. Defects are stored in the task/requirement management system, either as defects linked to user stories, or as comments on user stories, using the standard defect and story management practices and standard templates that have been configured into the tools. All agile teams must prepare a one-page Release Test Plan, Iteration Test Plans (whiteboard-based), Test Cases for high risk features, and an end of release one-page Test Summary Report.

Reporting: Test progress is reported verbally at morning stand-up. Due to client requirements, a point-in-time capture of test status is captured weekly on the team wiki, indicating numbers of tests passing, failing and blocked per sprint. In teams that are recording defects on separate defect cards, defect counts per severity level are also reported weekly on the wiki. Test and defect status are available real-time on the team's dashboards. Due to client requirements, all teams will be required to produce one-page Test Summary Reports.

Configuration management of test assets: The organization's requirement management, test management and test automation tools have in-built configuration management for all assets stored within each tool. All document-based assets will be stored on the team wiki, with manual version control applied to ensure each unique document version is captured and backed up.

Test levels, types & techniques: Types of testing that prove user stories meet their acceptance criteria and that reduce risks will be chosen during iteration zero and subsequent iteration planning sessions. Levels of testing typically applied include unit, integration, story/system, system integration, acceptance and production verification testing. Types of testing typically applied are functional, performance, accessibility, penetration and disaster/recovery testing. Entry criteria, exit criteria and test completion criteria are specified in the Definition of Done. Test techniques from ISO/IEC/IEEE 29119-4 are tailored to ensure testing remains lean.

Guidelines for deviation from Organizational Test Practices: Each team is required to use the same templates for requirement/story management, defect management and test management within the organization's suite of tools. Test documentation templates (i.e. for written documents) can be tailored as required by each delivery team.

Degree of independence: Each Agile team is assigned a tester that reports to the Delivery Lead. However, testers also have the ability to report any risks or issues to the Head of Testing (HoT), providing independence to all testing within the organization, as the HoT works in parallel to the Head of Development and reports directly to the Chief Technology Officer. The test organization is technically, managerially and financially independent from the development organization of Agile Corp, while within each agile team, assigned testers participate directly in self-organizing teams that report to their Delivery Lead.

Test environment & data: Developers conduct all unit and integration testing in the development environment. All testers conduct story/system and system integration testing in the system test environment. Users conduct acceptance testing in the pre-production environment. Performance, accessibility, penetration and disaster/recovery testing is also conducted in the pre-production environment. Production verification and penetration testing are conducted in the production environment. Obfuscated production data will be utilized during testing, ensuring all uniquely identifiable private data is removed or masked prior to testing.

Metrics:

— Test progress is measured according to team velocity, which is assessed at the end of each iteration

— Test automation coverage is measured by the percentage of acceptance criteria (from user stories) covered by automation tests, and is reported at the end of each iteration

From a product quality improvement perspective, the number of defects per severity level that slip through to production is assessed monthly, to identify opportunities to improve iteration and release testing processes.

Retesting and regression testing: Tests are automated wherever possible, reducing regression testing overheads. For all detected defects, an automated or manual regression test must be created.

D.2 Example 2 – Traditional Ltd

Organizational Test Practices – Traditional Ltd

Item	Practices
Risk management approach	All projects must conduct a product & project risk assessment based on the organizational Testing-Related Risk Management Process (TRM2019_04), which meets the requirements of ISO 31000 and utilizes quality criteria from ISO/IEC 25010 and ISO/IEC 25012. Risk assessments must be conducted at the start of each project, ideally while requirements are still being written, and revisited regularly (e.g. weekly) throughout each project. Any deviations from the specified risk management process must be approved by the Head of Testing.
Test selection and prioritization	Test cases and test procedures will be prioritized according to the risk associated with the requirements the cases are covering. If a test procedure includes test cases with different risk levels, the test case with the highest level determines the risk level for the entire procedure. Execution of test procedures must always be scheduled according to the risk, so that the higher the risk level the sooner the procedure is scheduled to be executed. Care must however be taken so that all feature sets are covered by some testing, that is no feature set must be left out of the execution schedule.
Test documentation and reporting	The test projects must be documented in such a way that an audit can establish what has been planned and what has been performed. Tracing between artefacts is essential. Test documentation requirements specified in ISO/IEC/IEEE 29119-3 must be produced throughout all projects.
Test automation and tools	The organization's central test management tool is to be used on all test projects and for all test levels and types. In the cases where more than 4 cycles of regression testing are planned, the project might consider automating testing.

Item	Practices
Configuration management of test work products	All test documentation must be stored in the central knowledge repository for each project (which is backed up and version controlled). All manual and automated test cases and execution results must be stored in the central test management tool. All automated test scripts, including for functional regression testing and performance testing, must be stored in the central source code control system.
Incident management	The company's Incident Management Process (IM2019_01) must be followed on all projects. This includes the mandatory requirement for documenting all defects or issues detected in each system.
Test levels	All projects must include the following test levels: — Static Testing: Requirement, Design & Code Inspections — Unit Testing — Integration Testing — System Testing — System Integration Testing — User Acceptance Testing — Production Verification Testing
Test types	All projects must consider applying the following types of testing, depending on the requirements and risks involved in each project: — Performance Testing — Disaster/Recovery Testing — Penetration Testing All other types of non-functional testing must be considered based on the unique requirements and risks involved in each project.
Guidelines for deviation from Organizational Test Practices	Deviations from the Organizational Test Practices will only be permitted with the express written approval of the Head of Testing and the Test Manager of the given project or program of work.

Item	Practices
Entry criteria	Entry Criteria for test execution: — Static Testing: Requirement, Design & Code Inspections — Artefact being inspected has been drafted — Unit & Integration Testing — Design & code have been implemented for a given feature — System Testing — Unit & Integration Testing are complete for the given feature — Test cases are designed, traced to requirements & signed off by stakeholders — System Integration Testing — Interface has been specified including functions & parameters — Internal or external system interface is available (either as real interface or stub) — Test cases are designed, traced to requirements & signed off by stakeholders — User Acceptance Testing — All previous stages of testing are complete and passed testing — Test cases are designed, traced to requirements & signed off by stakeholders — Production Verification Testing — All previous stages of testing are complete and passed testing — Test cases are designed, traced to requirements & signed off by stakeholders — Performance, Disaster/Recovery & Penetration Testing — Requirements for each type of testing have been specified and signed off by stakeholders — Test cases are designed, traced to requirements & signed off by stakeholders — Production-like environment is available and ready for testing — Required test data has been generated and is available For all test levels, it is recommended that a Project Test Plan (at a minimum) be signed off by the Project Manager, Test Manager and Head of Testing, prior to commencing each type of testing.