ISO/IE/IEEE 29119-3 Information Items	Normative requirement
iii) Priority	Shall
iv) Contents (Traceability)	Shall
b) Test procedures:	Shall
i) Unique identifier	Shall
ii) Objective	Shall
iii) Priority	Shall
iv) Start up	Shall
v) Test cases to be executed (Traceability)	Shall
vi) Relationship to other procedures	Shall
vii) Stop and wrap up	Shall
B.1.9 Test Data Requirements	Shall
a) Detailed test data requirements:	Shall
i) Unique identifier	Shall
ii) Description	Shall
iii) Responsibility	Shall
iv) Period needed	Shall
v) Resetting needs	Shall
vi) Archiving or disposal	Shall
B.1.10 Test Environment Requirements	Shall
a) Detailed test environment requirements:	Shall
i) Unique identifier	Shall
ii) Description	Shall
iii) Responsibility	Shall
iv) Period needed	Shall
B.1.11 Test Data Readiness Report	Shall
a) Test data status:	Shall
i) Unique identifier	Shall
ii) Description of status	Shall
B.1.12 Test Environment Readiness Report	Shall
a) Test environment status:	Shall
i) Unique identifier	Shall
ii) Description of status	Shall
B.1.13 Test Execution Log	Shall
a) Events:	Shall
i) Unique identifier	Shall
ii) Time	Shall
iii) Description	Shall
iv) Impact	Shall
B.1.14 Incident Report	Shall
a) Incident details:	Shall
i) Timing information	Shall
ii) Originator	Shall

ISO/IE/IEEE 29119-3 Information Items		Normative requirement
iii)	Context	Shall
iv)	Description of the incident	Shall
v)	Originator's assessment of severity	Shall
vi)	Originator's assessment of priority	Shall
vii)	Risk	Shall
viii)	Status of the incident	Shall

Annex C
(informative)

Overview of Examples

C.1 Overview

Annexes D to S contain examples of the application of the templates on both agile and traditional projects, to demonstrate the applicability of this standard to both types of projects. It should be noted that these are examples only, and many variations are possible and likely.

Particularly this is the case in agile projects. The reduced (more agile) information items presented in the agile examples are 'lightweight' versions of the information items. This approach is acceptable because of the lower perceived development risks, whereas the 'heavyweight' versions presented for the other examples are related to the higher needs for assurance in the lifecycle.

Any project can tailor documentation from full (all documents) to a minimal set of test documents, where minimal would be project defined.

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

The example documentation is based around two example projects:

Agile Corporation is a large publication organization producing magazines and books. The Corporation has an internal IT department, which is responsible for all the IT products that are in use by the organization, in supporting them in their business. Projects are run by a single agile team, so there are no projects performed using traditional development methods. The organization has several years of experience working in this way, and finds that it works really well with their needs for new and enhanced IT systems to support the business.

The project featured in this example is the development of a new web-based subscription system allowing people to become subscribers and allowing 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 critical, in the sense that wrong analysis results could cause prescription of wrong doses of fertilizer (either too much or too little). The organization is hence required to produce the product according to a specific standard that state requirements concerning production and quality assurance of certain documents and traceability between work product elements.

The project featured in this example is the development of the PC-part of a product called UV/TIT-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.

Not all of the example documents include the sections for Document Specific Information or Introduction; this is because this information is company-specific and the examples focus on the testing contents of the documents.

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 necessarily complete. Where paragraphs have been left out this is marked by three vertical dots, like this:

.
.
.

Omitted text is identified with an ellipsis, like this "**…**".

Annex D
(informative)

Test Policy

D.1 Example 1 – Agile Corporation

Agile Corporation is a large publication organization producing magazines and books. See more details in the introduction in Annex C.

Test Policy for Agile Corporation, V1.2 (02/13/2009)

Issued by: Ursula Mayers, Head of Development

Approved by: Stephan Blacksmith, Head of QA

Scope: This Test Policy describes the corporate view of testing for Agile Corporation and provides a framework within which all testing carried out on all internal projects within the organization will be performed.

Introduction: Agile Corp recognizes the need for testing of its internal products. The cost of developing high quality software systems can be considered as falling into four categories: prevention costs, testing costs, internal failure costs and external failure costs. It is generally cheaper to prevent defects than detect them and fix them (testing costs plus internal failure costs), and the highest cost is of external failure when detected by the users. To avoid this Agile Corp practices Test-Driven Development (TDD) and Acceptance Test-Driven development (ATDD), which are software development techniques. In its implementation of TDD, Agile Corp uses white-box techniques as defined in ISO/IEC/IEEE 29119 part 4.

Objectives of testing: The objective of testing is to provide sufficient information to determine the current quality of the system under test. As such all activities aimed at achieving this are considered to be software testing activities (e.g. integration, system, acceptance and regression testing).

Test process: The software testing will be based on test processes as defined in ISO/IEC/IEEE 29119-2 and aligned with the development approach.

Test organization structure: Testing will be resourced within the Agile Corp from a central pool of testers who are assigned to the project. In addition, a central 'expert' software testing resource led by the Head of Testing will provide test consultancy services to projects as necessary. Test organization structure inside of a project will follow project guidance.

Tester training: All members of testing teams are expected to have appropriate university education or at least a minimum level of industry certification in software testing. Additionally, testers are expected to be knowledgeable in agile concepts, or to become so within three months of joining a test team.

Standards: Test documentation will be based on ISO/IEC/IEEE standard 29119-3 "Test Documentation", adapted for use in agile projects.

Other relevant policies: Software Development Policy for Agile Corporation, V4.3 (12/12/2008)

Test process improvement and value determination: End of iteration retrospectives will capture lessons learned, metrics, and improvement concepts which will be provided to the central test organization.

D.2 Example 2 – Traditional Ltd

Traditional Ltd is a small company that produces advanced analysis equipment for the farming industry. See more details in the introduction in Annex C.

This policy is published on the Traditional intranet under Management >> Policies. Hence it does not contain all the document related information, and it is not versioned, but a publishing date is visible.

Test Policy

Objective and definition of test

At Traditional Ltd, testing is considered as a means to achieving user and customer confidence in our products. Testing is one of many means to achieve this goal.

Testing process

Any software project must include a test project. In other words, the test project must be a subproject of a corresponding software project.

The two projects should be started at the same time. The test process includes the activities: planning, analysis and design of test material, execution and recording of the test including registration of any incidents, and test completion and reporting. Testing is affecting something (object under test), observing the effect and deciding whether this effect is considered correct or incorrect behaviour.

Organization

Each project will be staffed by analysts, designers, programmers, and test analysts. They will all report to the project manager. Students could be hired to execute tests.

Evaluation of testing

For each product, management should decide which level of quality is to be achieved expressed as the maximum number of incident reports over a given period from customers.

At the release of a product the test group must deliver a report on the product's expected behaviour. One year after release, management reviews this report and compares it with respect to feedback from the market (number of incident reports, number of failures.)

Standards

We follow our own standards found on the intranet. These are all based on ISO/IEC/IEEE standard 29119-3 "Test Documentation".

Policies in Traditional Ltd

The policies for Software Development and Quality Assurance form the basis for all software development and testing in Traditional Ltd.

Approach to test process improvement

At the release of a product the test group must deliver a report which analyses the project from a testing point of view. Any improvements suggested in this report are discussed with management in order to decide which improvements to make.

When the evaluation of the testing takes place one year after release, management considers if improvements should take place.

Annex E
(informative)

Organizational Test Strategy

E.1 Example 1 – Agile Corporation

Agile Corporation is a large publication organisation producing magazines and books. See more details in the introduction in Annex C.

Organizational Test Strategy for Agile Corporation, V1.1 (03/23/2009)

Issued by: Ursula Mayers, Head of Development

Approved by: Stephan Blacksmith, Head of QA

Issuing organization: The head of testing at Agile Corporation is responsible for preparing the Organizational Test Strategy. Upon review and approval, the senior management at Agile Corporation is responsible for distributing the Organizational Test Strategy.

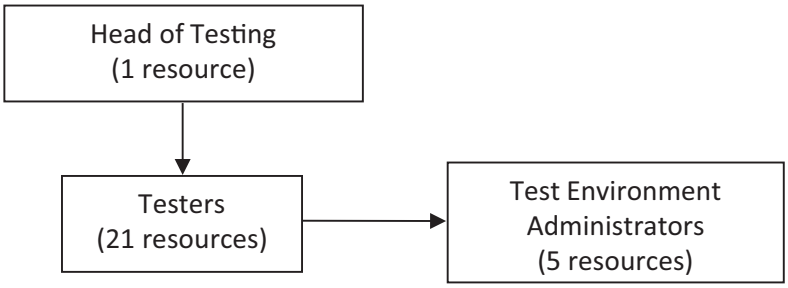
Scope: This Organizational Test Strategy provides the Corporation's overall approach to testing. We have developed and implemented several guidelines that are applicable across all projects. We aim to provide testing at every point in the system and software lifecycle. This is accomplished by having our test group engaged early in the lifecycle process, participating in teams with developers, and working with user stories even in a draft state. These pertinent artefacts provide the basis for establishing test plans and scoping the test effort. In addition to the development of test plans, the organization will use agile test activities such as stakeholder involvement in test design, preparing test automation, peer reviews, various testing design techniques (project applicable), lightweight defect tracking and reporting.

References: Agile Manifesto

Generic risk management: All risk management must follow the prescribed corporate Risk Management process as defined in Corporate Policy-RM56, where the general risk register is identified. Any deviations and waivers must be approved by senior management.

Degree of independence: The Corporation test organization is led by the head of testing who has no direct association with the head of development. The test organization is technically, managerially, and financially independent from the development organization of Agile Corp, while within a project assigned testers could participate directly in self-organized teams, which include development.

Test organization structure: Agile Corporation Test Organization has a pool of independent test professionals from which testers are assigned to agile teams, e.g. a scrum team, where testers are members of the overall team.



Test documentation strategy: The Test Organization will comply with the test documentation defined in ISO/IEC/IEEE 29119-3, and the principles of agile development. Any deviations will require the approval of the head of testing.

Test sub-processes documented in project test plans: The Corporation relies on our highly competent head of testing to help ensure the most effective type of testing is invoked. This is accomplished via a mentoring program with the testers on scrum teams, and it includes functional and non-functional methods, test design techniques, and testing tools, tailored from ISO/IEC/IEEE 29119-1, -2, and -4. Additionally each project defines test selection, priority and management. Further, the project must select their own test environments, retest/regression practices, and incident management practices. These items are agreed during continuous direct interaction with stakeholders over the life of each project. The level of test plan documentation (size and format) is also agreed with project stakeholders.

E.2 Example 2 – Traditional Ltd

Traditional Ltd is a small company that produces advanced analysis equipment to the farming industry. See more details in the introduction in Annex C.

Traditional Ltd has an organizational test strategy with a project-wide part, and a part for each test sub-process. This example only includes the project-wide part and the parts for component test and system test.

Organizational Test Strategy

Organizational Test Strategy	
Issue	**Strategy**
Generic risk management	The risk management in a project must be based on the generic risk register for the specific type of project and the generic risk management process. The risk registers are found in XX. When a project is closing down the relevant generic risk register(s) must be updated as appropriate.
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. A project test plan and a project test completion report as outlined in ISO/IEC/IEEE 29119 Part 3 must be produced at the test project level.
Test automation and tools	The test management tool BCG is to be used on all test projects and for all sub-processes. In the cases where more than 4 regression tests are planned the project might consider using a capture/playback testing tool.
Configuration management of test work products	The Traditional Ltd process for configuration management must be followed for all test work products.
Incident management	The Traditional Ltd process for incident management must be followed.
Test sub-processes	Each test project must include the following test sub-processes: - Performance test – if applicable in relation to requirements - Operability test - Component testing - Component integration testing – preferably bottom-up - System testing

Organizational System Testing Strategy	
Issue	**Strategy**
Entry and Exit criteria	The integration test completion report and the system test specification must be approved before the system test execution might begin. All the system test deliverables must be approved before the system test is finished.
Test completion criteria	The system test is supposed to achieve 100 % requirements coverage, and all test procedures must be executed without incidents.
Test documentation	A system test plan and a system test completion report as outlined in ISO/IEC/IEEE 29119 Part 3 must be produced, as must all documents defined for dynamic testing.
Degree of independence	The system test must be specified by the staff in the test department and executed by students.
Test design techniques	Appropriate black-box test case design techniques are to be used. Error guessing could also be used, if defect information exists for previous versions.
Test environment	The system testing environment must be identical to the production environment in terms of hardware and software. In the case of embedded systems the system test could be executed on a simulator. Data could be made anonymous, but must otherwise be 100 % representative.
Metrics to be collected	The following shall be reported in the system test completion report: - Total number of specified test procedures - Total number of executed test procedures - Total number of testing hours spent on specification - Total number of hours spent on execution and registration of incidents - Total number of hours elapsed for testing - Total number of failures found
Retesting and regression testing	All test procedures resulting in incident reports must be rerun after defect correction. Regression testing during the system test sub-process is at the test manager's discretion. In the final system test run, all test procedures must be executed.

Organizational Component Testing Strategy	
Issue	**Strategy**
Entry and Exit criteria	The test item (component) must compile and link, and the component test specification must be approved before the component test execution can begin. All the component test deliverables must be approved before the component test is finished.
Test completion criteria	The tests for each component is supposed to achieve at least 90 % statement coverage and at least 80% decision outcome coverage, and all test cases for a component must be executed without incidents. Reason for non-conformances must be reported and accepted by the project manager.
Test documentation	A component test plan and a component test completion report as outlined in ISO/IEC/IEEE 29119 Part 3 must be produced; Test cases to help ensure coverage must be produced for each component.
Degree of independence	The component test must be specified and executed as a peer test, that is, by a developer who is not the one who coded the component under test.
Test design techniques	Appropriate black box test case design techniques are to be used, and these must be supplemented by the white box techniques: statement testing, and decision outcome testing, where necessary to help ensure required coverage.
Test environment	The component testing can be executed in the development environment of the developer who is designing the test, i.e. not the environment of the developer who coded the component under test.
Metrics to be collected	The following must be reported in the component test completion report: • Average obtained statement coverage. • Average obtained decision outcome coverage. • Total number of incidents found and corrected.
Retesting and regression testing	Each component must be retested until completion criteria have been reached.