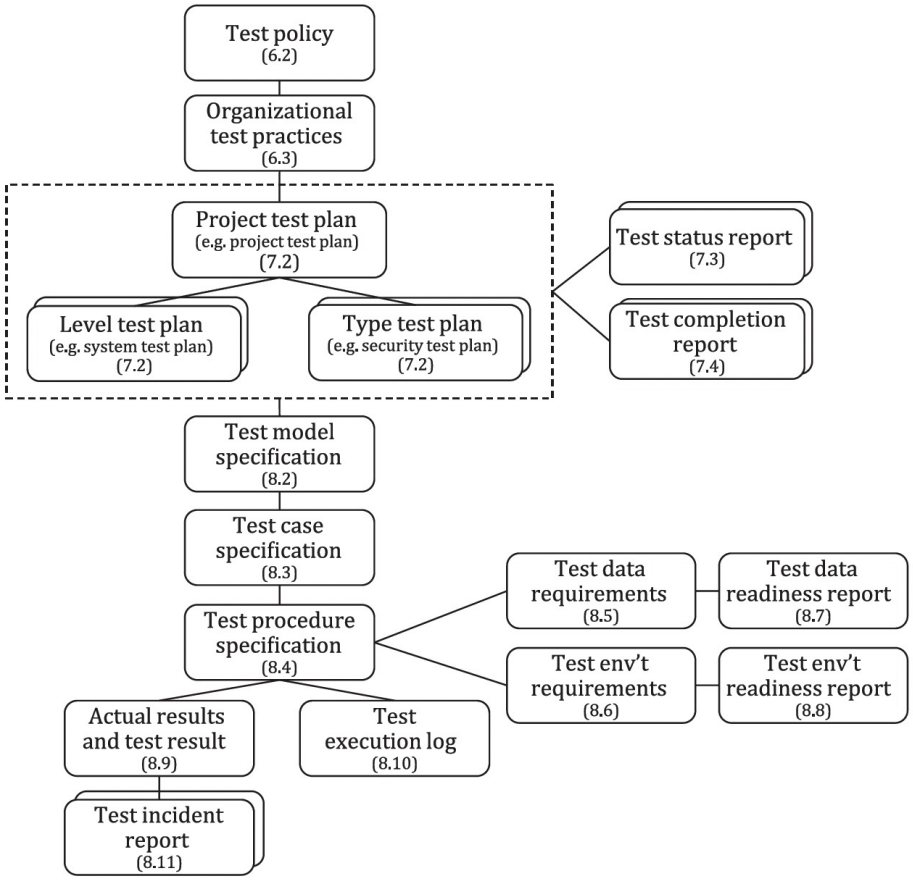
ISO 29119-3

Overview

- ISO 29119-3 provides two examples of test processes:
  - A traditional approach based on a waterfall-type process that is fully conformant to the standard.
  - An agile process that provides an example of a “tailored” approach to conformance where the decision on what to include is risk-driven.
- We will look at the agile process – please look at the traditional process
- The standard specifies the main elements of test documentation and what they should record:
  - *6. Organizational test process documentation*
  - *7. Test management processes documentation*
  - *8 Dynamic test processes documentation*

Overview of test documentation



Example Process: Traditional

- *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.*

Example Process: Agile

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.

Test Policy: Agile

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

Test Policy: Agile

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.

Organizational Test Practices: Agile

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.

Organizational Test Practices: Agile

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.

Organizational Test Practices: Agile

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.

Organizational Test Practices: Agile

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.

7. Test management processes documentation

7.2 Test Plan

- Context of Testing
- Assumptions and Constraints
- Stakeholders
- Testing Communication
- Risk Register
- Test Strategy
- Testing activities and estimates
- Staffing
- Schedule

7.3 Test Status Report

- Test status
- Reporting Period
- Progress against test plan
- Factors blocking progress
- Test measures
- New and changed risks
- Planned Testing

7. Test management processes documentation

7.3 Test Completion Report

- Summary of testing performed
- Deviations from planned testing
- Test completion evaluation
- Factors that blocked progress
- Test measures
- Residual risks
- Test deliverables
- Reusable test assets
- Lessons learned

Summary

- Section 7 of ISO 29119-3 covers the main test process documentation
- These documents are described in generic terms
- In appendices E-G these are elaborated relative to a traditional and agile lifecycle
- What follows is a brief summary for the agile process

Test Plan: Agile

- **Context of Testing:** Keep test types and levels on the story cards stored in the company requirements repo.
- **Assumptions and Constraints**
- **Stakeholders:** Product owner and delivery lead are the key stakeholders for testing, includes escalation to more senior levels.
- **Testing Communication:** Lists the team: Delivery lead, Developers, Business analysts, User representatives, Testers, Product owners plus reporting arrangements
- **Risk Register:** Risks recorded on story cards and wider risk register – methodology probably follows a standard such as ISO 16085 or NIST Risk Management Framework
- **Testing activities and estimates, Staffing, Schedule** follow the agile process and strategy determined at strategy level

Test Plan: Agile

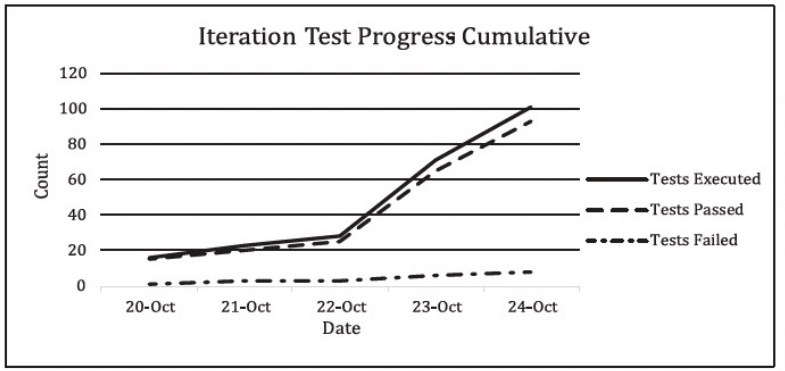
- **Test Strategy**
  - At start of iteration estimate the necessary effort including any backlog
  - Use Test Driven Design to create and automate unit and integration tests in advance of coding, create system and acceptance tests
  - Regression test features that may have been influenced by changes
  - Design tests with a view to easing acceptance and taking risk of feature failure into account
  - To mark a story DONE there must be no fatal or severe defects, remaining defects get made into new stories and added to the backlog
  - Outline progress and identify blockers at daily stand-up meetings
  - Ensure tests, scripts etc are available in the central repo
  - Issue short summary of testing outcome at the end of iteration

Test Status Report: Agile

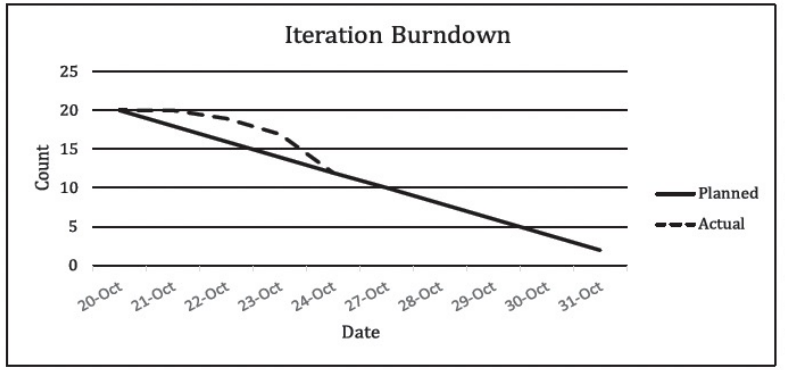
- **Test status:** Weekly report on the team wiki
- **Reporting Period:** The week this refers to.
- **Progress against test plan:** Report against the user stories in play e.g. 8 DONE, 2 still open, focus on high risk stories and give coverage data, give profile of the risks at different severity, report on automation progress.
- **Factors blocking progress:** things like delay caused by UI changes, how these were resolved, e.g. extra effort, …
- **Test measures:** sessions sheets developed for new stories with some assessment of test ideas on those sheets.
- **New and changed risks:** report on any new or modified risks
- **Planned Testing:** activities planned in the coming week

Test Status Report: Agile - measurement

Iteration Test Progress - Cumulative					
Date	20-Oct	21-Oct	22-Oct	23-Oct	24-Oct
Tests Executed	16	23	28	71	101
Tests Passed	15	20	25	65	93
Tests Failed	1	3	3	6	8



Iteration Burndown (Iteration 3, Week 1)					
Date	20-Oct	21-Oct	22-Oct	23-Oct	24-Oct
Planned DONE	2	2	2	2	2
Actual DONE	2	0	1	2	5



Test Completion Report: Agile

- **Summary of testing performed:** summarise what the report covers – usually a release of the software
- **Deviations from planned testing:** this is covered by the agile process as the testing evolves in line with the needs of developers in TDD.
- **Test completion evaluation:** to get to a release all the high-risk stories are marked DONE and residual risks accepted by the product owner
- **Factors that blocked progress:** these will have been recorded in the log of the previous iterations
- **Test measures:** these are built into the agile process
- **Residual risks:** kept on the story cards and wider risk register all have been accepted by the product owner

Test Completion Report: Agile

- **Test deliverables:**
  - Report on the stories that are in play for the release, all must have passed the system integration testing (SIT) and the user acceptance testing (UAT)
  - Measures of quality of the code, coverage measures, tests passed etc
  - Report on remaining defects and how they have been moved forward into the next iteration (new stories).
  - Report on product owner and user approval
  - Other features, like accessibility testing, resource use, ethics etc.
- **Reusable test assets:** built into the agile process tests are kept in the repo and can be reused and reworked
- **Lessons learned:** suggestions to improve resourcing and resource estimation, suggestions for innovation in test techniques, identify the need for data from users (including necessary privacy preserving measures), …

Summary

- Please read Appendices C-G of ISO 29119 – in particular the traditional approach the covers strict conformance using a traditional lifecycle.
- This slideset covers the incorporation of test planning into an agile development process.
- The example is covered in more detail in the standard.