**‘BREAK’ SOFTWARE TESTING TEST POLICY TEMPLATE draft v.02**

This template is intended to give some insights in what is important in a software testing policy.  
Use it to create your own more sec business look document.   
This is mainly for inspirational purposes and to make you think about topics.  
Complete English translation is further down.

**Section 0: For AGILE (SCRUM)**

Clarify and document whether story points include or exclude software testing.

|  |  |
| --- | --- |
| A close up of a sphere  Description automatically generated   * Story points **exclude** testing. | A blue and white sphere  Description automatically generated  Story points **include** testing. |

Decide, define, and document this explicitly.

Is there a field in the project management software's story item type (e.g., in Jira, used for bug tracking and issue tracking)? If so, what are the rules for when and how this field is filled?

A screenshot of a computer

Description automatically generated

For example, if there is a (custom) field called *‘Test Preparations’*, when is this field completed, and with what information? What is the precise meaning of *‘Test Preparations’*?

If *‘Test Preparations’* represents test effort (e.g., High/Medium/Low), what do these indicators mean exactly?

**Section 1: Responsibilities**

This section must address the following:

* Define whether software testing is **decisive** or **advisory** regarding production readiness. Advisory is preferred.
* Clearly outline what happens if the software is not ready for production.
* Specify the criteria for moving to production or to the next phase in the OTAP (Development, Testing, Acceptance, Production) cycle.

Options:

|  |  |
| --- | --- |
| A person standing in front of a group of people around a table  Description automatically generated   * The software tester **advises**. | A group of people sitting at a desk  Description automatically generated   * The software tester **decides and acts**. |

Clarify whether, if a software tester advises against production, someone in the organization can override this decision and move forward with the release. If so, who can do this, and how is it decided? This must be defined in this section.  
  
Related reads: [How to introduce QA in a company? [part1] - Testenvansoftware.nl](https://testenvansoftware.weebly.com/software-testen---blogs/how-to-introduce-qa-in-a-company)  
[How to introduce QA in a company? [part2] - Testenvansoftware.nl](https://testenvansoftware.weebly.com/software-testen---blogs/how-to-introduce-qa-in-a-company-part2)

**Section 2: Test Reporting**

Describe the format and content of a test report.

* What should it include?
* Should it cover results from functional testing (manual), acceptance testing, user acceptance testing, regression testing, **and** automated test outcomes?

Clarify the expectations for the test report.

A computer keyboard and a machine

Description automatically generated with medium confidence

**Section 3: Testing Layers**

Define which testing layers are reported on. For example, in the testing pyramid, which layers are included?

* Are component tests included, or just unit tests, integration tests, and E2E tests?

Example: Does the final (test) report consist of:

A diagram of a layering system

Description automatically generated

If not, describe the correct approach here.  
  
Related reads: [Understanding software testing a bit better. - Testenvansoftware.nl](https://testenvansoftware.weebly.com/software-testen---blogs/understanding-software-testing-a-bit-better)

**Section: Test Plan**

Define when the *Test Plan* field is filled and what it contains.

* Related reads: [Test process in details. - Testenvansoftware.nl](https://testenvansoftware.weebly.com/software-testen---blogs/test-process-in-details)
* [Real software testing is far more difficult then most people think... - Testenvansoftware.nl](https://testenvansoftware.weebly.com/software-testen---blogs/real-software-testing-is-far-more-difficult-then-most-people-think)
* [Software testing insights - part3](https://www.youtube.com/watch?v=qmkYIk6x0o8&t=185s)

**Section: Requirement Traceability**

One of the most critical aspects is requirement traceability. For example, if an automated E2E test fails in May 2030, it should be clear why the test failed and whether this constitutes a defect.

Clarify which requirements exist, where they are located, and how they are covered by tests.  
A diagram of scrum software

Description automatically generated

Related reads: [Software testing insights - part4 - YouTube](https://www.youtube.com/watch?v=FZ2X5HSv4K4&t=2s)

**Section: Test Design Techniques**

Define whether and how test design techniques are used.

* When are they expected?
* When are they applied?

What is a test design technique in the context of software testing?

**Section: Requirements Leadership**

Define what is authoritative to avoid misunderstandings.

* A statement like, *“Anja from administration said it should work this way”* is **not** a requirement.

So, when does something **become** a requirement?

* If a requirement is incorrectly documented but still tested (and approved), what prevails?

If written requirements are binding, determine where they are expected to be documented and when they are considered final (e.g., when a story enters the test lane).

How are requirement changes managed?  
  
Related reads: [ACCREQ administration in software testing - Testenvansoftware.nl](https://testenvansoftware.weebly.com/software-testen---blogs/2-maanden-geleden)

**Section: Test Environments**

Define how test environments are managed.

* It is challenging to work in an environment undergoing maintenance. What are the agreements regarding this?

A person painting a statue

Description automatically generated

**Avoid "shooting at a moving target" if possible** (except perhaps when testing in production). A test environment should be stable.

* Define if and how a code freeze is implemented.

A person on a horse with an object and a bird flying in the sky

Description automatically generated

**Section: Test Data**

Describe how test data is managed.  
  
How is the Data of the OTAP managed? How much, why, when may they differ from environment to environment?  
The intention is to keep the environments as similar as possible.

**Section: Artificial Intelligence in Testing**

Describe whether and how Artificial Intelligence can be used in software testing.

* Related reads: [blogs about software testing - Testenvansoftware.nl](https://testenvansoftware.weebly.com/software-testen---blogs/ai-is-nog-lang-niet-foutloos)

**Section: Testing Priorities**

Clarify how testing priorities are determined.

* Is a Product Risk Analysis (PRA) conducted?
* Are priorities determined per story? If so, how does this impact regression testing?

What determines testing priorities? Define this in this section.

**Chronological Process Flow**

Describe the process flow for all testing activities. [Chronological Process Flow](https://softwaretestingbreak.com/processflow.html)

More info about the methodology: [A screenshot of a computer

Description automatically generated](https://softwaretestingbreak.com/)