Test Strategy

Version 1.0
Date: 23/10-2025
MToGo

Introduction & Scope

As a company providing service-based solutions, this document describes our strategy for ensuring the quality of our products. It defines what a project based test plan should contain for maintaining standardized product quality, key points of concern, testing priorities, responsibilities and documentation.

In Scope:

- In-house developed application logic
- Internal services
- Integration with internal services
- Integration with External services
- Project relevant performance/stress tests

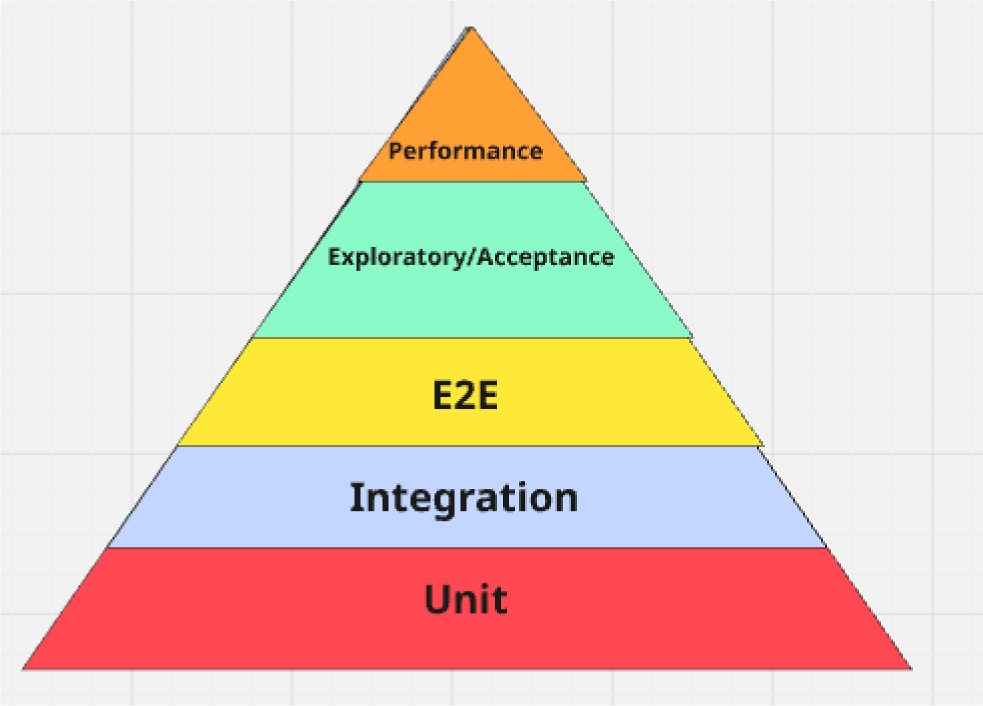
Test Objectives

- Sufficient test coverage with expected behaviors determined by testing tools.
- Determining which integration tests to be performed
- Identifying performance bottlenecks using performance testing
- Prevent functional regressions
- Code practice enforcement

Testing Levels

Our testing levels represent the test triangle containing unit, integration and system levels. Underneath is the priority of our tests, with the first mentioned being the tests we have the most of.

1. Unit testing application logic on individual class level.
2. Integration testing.
3. E2E service, and validation.
4. Exploratory and acceptance test.
5. Performance testing.



Testing Types

Functional: Application logic
Regression: Only adding to functional tests
E2E: Testing isolated behaviour of a service
Style checking: Reject code not living up to code standards

Roles and Responsibilities

Devs: Writes all of the tests. They run them locally on dev containers. Trigger automated runs on remote pushes before commit integration. The developers have a lot of responsibility and are both part of the dev and ops team, to fully integrate them into all parts of code production. Devs have the responsibility of making sure their code production and tests are of high quality.

Tech Lead: Creates and validates test plan in cooperation with devs, QA and optionally customers/partners.

QA Lead: QA Lead has the responsibility of code reviewing each week with the devs, to make sure the code is of high enough quality, working as expected.
Furthermore the QA Lead reviews the results of the weekly Load testing, determining if they meet the standards set.

Operations: Deals with tasks regarding server maintenance, health checks and monitoring of all systems. This includes creation of new servers, roles, DB setup, image handling, scaling and network management, etc.

Product owner: PO handles accept criteria validation and DoD. They need to confirm previous work and provide continuous feedback, to ensure the product is heading in the desired direction.

Security expert: Is charged with monitoring potential security risks and ensuring that OWASP standards are met.

Entry and Exit Criteria

Unit and integration testing

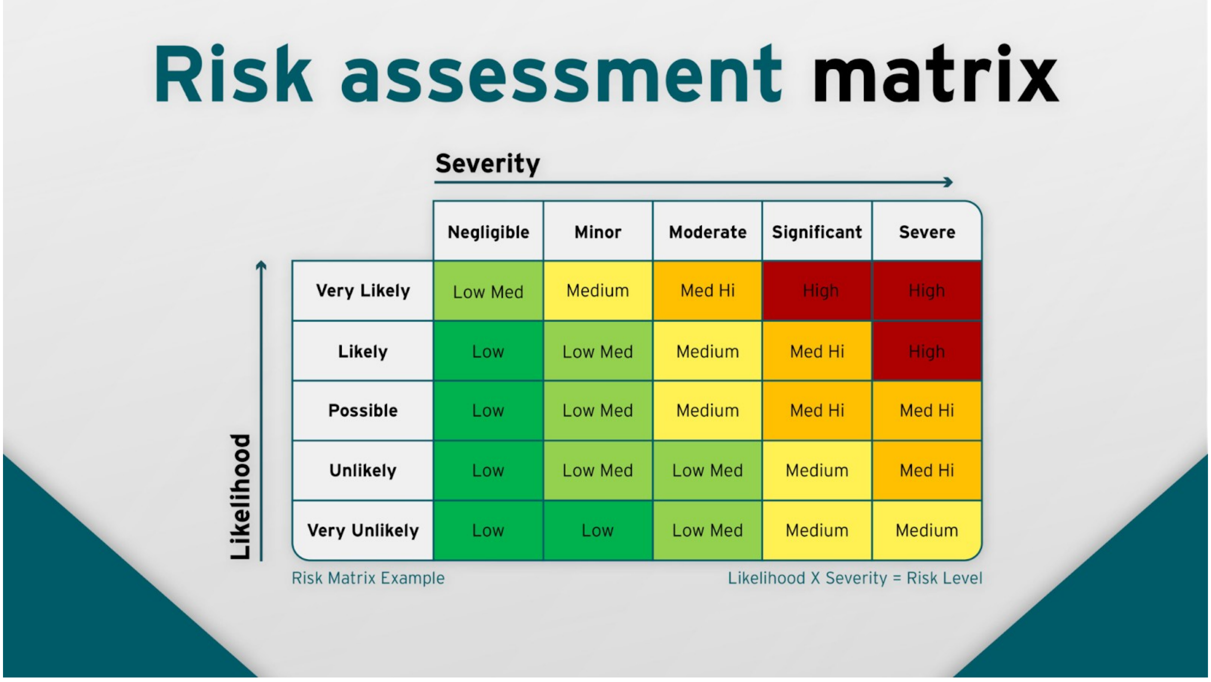
Entry: CI test environments set up
Entry: Local test environments set up

Exit: Functional prototype running in test env. All test cases executed

Performance testing

Entry: Build with vertically sliced functionality running in test env
Exit: Satisfactory system behavior including documentation during testing in test env.

Risk and Mitigation



Risk assessment matrix

Severity
Likelihood

	Negligible	Minor	Moderate	Significant	Severe
Very Likely	Low Med	Medium	Med Hi	High	High
Likely	Low	Low Med	Medium	Med Hi	High
Possible	Low	Low Med	Medium	Med Hi	Med Hi
Unlikely	Low	Low Med	Low Med	Medium	Med Hi
Very Unlikely	Low	Low	Low Med	Medium	Medium

Risk Matrix Example

Likelihood X Severity = Risk Level

	Risk	**Mitigation Strategy**	**Risk level**
1	Server breakdowns and network partitions	Quickly deployable application images, and network partition tolerant database backups	Med High
2	Cybersecurity attacks and breaches	Basic OWASP compliant systems	Med High
3	App crashes on exceptions/errors	Error tolerant unit and integration tests	Low Med
4	Bugs being pushed to production	Quick bug fixing/Optimizing since the company works trunk based for all projects	Low Med
5	Team member is sick	Unload tasks to other team members if they are in the position to and the tasks are of importance.	Low

Test Deliverables

- Unit test reports
- Code coverage reports
- Mutation test reports
- Performance reports
- Code style reports