# Definition of Done

Unit tests:

* Test Execution: All unit tests must be written and executed successfully without any failures.
* Unit test coverage should be thorough, covering critical parts of the codebase to catch potential issues (example: core features such as bulk scraping). Small functions don’t necessarily need unit tests.

Peer Review:

* Pull requests are reviewed by at least 2 other students

Documentation:

* Software requirements: Document clear and concise instructions on the software requirements necessary to install and run the code. Include specific versions, or dependencies.
* Source code documentation: Ensure that source code is well-documented with inline comments and clear code structure. A README file (for each component or core feature if needed) should provide an overview, its purpose, and how to set it up and use it.
* API/CLI Documentation: Provide detailed documentation for APIs and command-line interfaces (CLI). Include endpoints, input/output formats, usage examples and customizability.
* Maintenance Guide: Create a maintenance guide that explains how to install, update, and maintain the software. Include instructions on checking logs for debugging and identify which files to modify for specific tasks.
* General Software Documentation: Develop user flow diagrams and sequence diagrams to illustrate the software's functionality and interactions. These diagrams should help users and developers understand the software's behavior and logic.

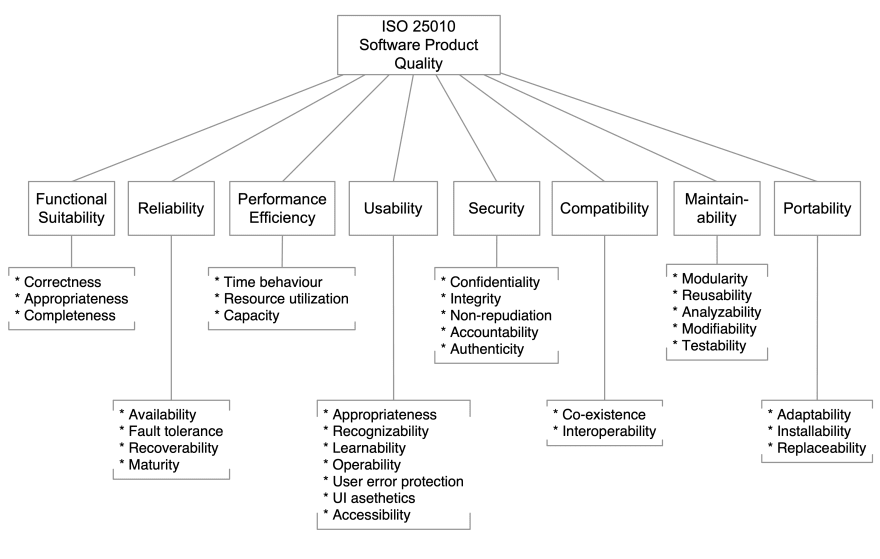
Use case fully working:

* TPE analyst (end-user) using the UI tool to scrape and analyze a website.

**Acceptance criteria:**

* Hands-on demo with PO at least once per sprint. Features are accepted as partially complete or complete by the PO based on above definition and the following quality attributes.
* Multiple milestones with TPE analyst to use the code. Analysts will be tasked with using the tool without any direct interaction with the students as tests of the documentation and usability. This will also generate feedback for improvements.

# Quality attributes

We use the ISO-25010 quality categories:

We have defined the relevant attributes and sub-characteristics for different components of the project along with metrics for a selection of sub-characteristics.

**Overall system**

* Functional Suitability: Correctness
* Maintainability: Modularity
  + Library has 7 modules
* Functional Suitability: Completeness
  + Evaluated as complete from F-Secure in final sprint review.

**User interaction**

* Usability: Operability
* Usability: User error protection
* Usability: Learnability
  + 4 cli commands with, on average, 14 documented options.
* Usability: UI aesthetics

**Code and API**

* Maintainability: Modifiability
* Maintainability: Testability
  + Code test coverage percentage from pytest: 78.04 %
* Portability: Installability
  + Minutes for a new developer to setup the project with Docker: 9 min
* Reliability: Fault tolerance
  + How long does wrong API call returns and not get passed through: 0.1 s
  + How long does it take a correct API call to return a result: 11 s

(1m 19s for 12 websites, 6 s/site)

**Scrapping and extraction functionality**

* Reliability: Maturity
* Performance Efficiency: Time behaviour
  + How long does the bulk scraping take to complete:
    - Run using AMD Ryzen 7 5800X 8-Core 3.80 GHz processor. Software was run inside VirtualBox with 16 GB RAM.
      * 100 sites, playwright: 2:44 min
      * 100 sites, curl: 0:16 min (most sites return very little data if unrendered)
* Performance Efficiency: Resource utilisation
  + Run using AMD Ryzen 7 5800X 8-Core 3.80 GHz processor. Software was run inside VirtualBox with 16 GB RAM.
    - Average CPU load to run scraping: Playwright for 100 sites: ~50% (not accurate)
    - Memory consumption: not measured
* Maintainability: Modifiability
* Reliability: Fault tolerance
  + Playwright scrape with top 100 urls (same as above) did not return errors or null-bodies.
  + Playwright scrape with top 1000 urls got 3 unrecovered errors, 4 time-out errors, 1 network error, 46 null-bodies.

**ML functionality**

* Maintainability: Modularity
  + The feature extraction and ML are separate modules with well defined input and output so can easily be developed independently.
  + Feature extraction module can be used for other purposes, not only as input for fake shop classification.
* Maintainability: Analysability
  + All features are documented with multi-line and inline comments
* Maintainability: Modifiability
  + Time to retrain a model with a new data set: <1 hour
  + Time to retrain a model with a new feature (after feature is ready): <1 hour
  + Time to make a new feature: 1 hour - 2 days