

# TULON.CZ'S TEST CLASSES CHEAT SHEET

## A. Positive tests

1. Smoke tests (bare minimum required for even attempting more tests)
2. Inputs that should be included in output successfully
3. Industry-standard values based on usage profiles
4. Consistency of processing

## B. Negative tests

1. Data/functional (against false positives, false negatives)
2. Failure Mode tests (error and exception handling)
3. Fail-safe (if justified by SW malfunction impact/Integrity level)

## C. Advanced & Destructive tests

1. Advanced/atypical activity handling tests
  - a) Scenario/Stateful testing (sequence of specific steps results in unique SUT state)
  - b) Using allowed but rare utilities and options
  - c) Rollbacks and Incident Recovery
  - d) Concurrent activity (eg. SUT run on some data while data being modified by utility)
  - e) State Transition testing (back-and-forth changes, unexpected from-to transitions)
2. Advanced/atypical objects handling tests
  - a) Using allowed but rare object parameters, variants and options
  - b) Advanced objects (DB table's associated Clone table, exotic XML, containers)
3. Advanced/atypical system states handling tests
  - a) Once-in-a-nevermind scenarios (leap year, „2K“, log rotation, DST changes, etc.)
  - b) Host system in degraded mode/limited functionality
4. Fault Insertion Tests (data corruption, incompletely written configs etc.)
5. Other relevant Failure Modes (what RCA, FTA, FMEA identified and didn't fit elsewhere)

## D. Equivalence partition testing

1. All boundary values (including just above, below, and on each limit)
  - a) Equivalence Partitions handling (all options + invalid choice processing)
  - b) Freeform input value length limits (including multi-lines when applicable)
  - c) Internal variables limits and specialities (against underflow, overflow, zero...)
2. Correctly processing pre-defined switches/options (path testing)
3. Stress testing / Load limits testing (maximum allowed items Nr., response near SUT limits)

## E. Input tests

1. Human Operator Error tests
  - a) Input sanitization handling
  - b) Character conversion handling
2. Internal data definitions handling
3. External inputs handling (object names, object lists, load files, config files)

## F. Output tests

1. Actual SW payload (correctness tested in TCI 1/2/3, here focus on formatting, paging etc.)
2. Messages and codes shown to the user
3. Correct file/dataset handling (allocation, de-allocation, file/FS consistency, etc.)
4. Databases interaction (not only payload, but also logging, saving timestamps etc.)

## G. Feature/implementation specific tests

1. Compliance to government regulations (EU Directives/Regulations; US FCC, U.S.C.; etc.)
2. Conformance to relevant industry standards (RFC, IEC, IEEE, ISO, ANSI, W3C, etc.)
3. Interaction with utilities or maintenance (handling DB REORGs; FS Defrag; etc.)

## H. User front-end tests

1. GUI tests
2. Help tests (Help panels, help files, guidance messages and pop-ups, tooltips, etc.)
3. Device/terminal compatibility testing (terminal/screen sizes, font availability, alien OS, etc.)

## I. System integration and interoperability tests

1. Interdependency on internal features and systems (including Unexpected Input Errors)
2. Interoperability with external services and systems (both inputs from and outputs to)
3. Interference from external services and systems (can unrelated systems cause us troubles?)

## J. Performance testing