Webtop and Webtop-Mx connector rough test plan

**Goal:**

1. Automation test case development
2. Cover Webtop APIs by cases
3. Manage scripts and maintain codes depositary

**Scope:**

Phase 1. Sanity check on all Apps (*expectation is unsure*)

[Cover ] All APIs and check response for:

1. Check single API - Not return error message
2. Check single API - Response include keywords(or value) according to detail spec
3. Check function – API successfully send also function works

Phase 2. Full regression test on all APIs

[Cover]

1. attributes (variety attributes )
2. attributes data – if value is discrete and limited data, cover all, if not discrete, cover classic variety data
3. cover special data - API works for escaped special data

Phase 3. Jenkins link for test on server, maintain codes

**Method:**

1. Base on real product build
2. Develop cases locally and manage on Bitbucket
3. Sanity test is focus on single API (or if need, step by step like manual test logic to cover as many APIs as possible in one case)
4. Full function test is focus on API attributes.

**Currently we have:**

1. Robotframework as the test cases driven tool
2. The 3rd party python package(requests) used for http transfer
3. Code structure and source code management depositary
4. Jenkins running environment
5. Test environment for Webtop API: Kiwi2.0
6. APIs grabbed from UI, services lib for basic functions

**What we don’t have:**

1. Get the detailed webtop api documentation (if developer can’t give such list, we need manually search all apis, see point2)
2. Complete API specification in order to assert the response in the right way
3. Case logic standard: Individual API or manual user scenario

**Rough estimation:**

Depends on the mount of webtop APIs we need to cover.

**Risk**

1. Scope: Features and functions not sure about Mercury, OTOSAN and OCTANE (if need to cover)
2. New features and updated API for different projects
3. API data update, data format update (like XML update to Json)
4. Will not cover front-end like special characters escape