

# Test Strategy

A test activity can be aimed at verifying the software system based on specific reason or target for testing. That is why the tests should be different.

Here are the supposed classification of tests, which we need to be sure that the product is qualitative.

We should have the types of tests, described below (not only Unit tests, but also Integration tests). Each test in Jira should be marked by specific label, which reflects the relation of test to the specific test type. Tests can be a marked by several labels.

## 1. **UI testing** (label – “UI”).

That means that we need to verify all pages of Kronos, Pegasus, Rhea, Apollo, xConnect portals and API interfaces by these tests, which includes:

- the appearance of pages, according to mock-ups and requirements;
- the validation of forms (positive as well as negative scenarios);
- the work of buttons, tabs;
- the functionality of links, hyperlinks, bread crumbs;
- filters and sorting.

## 2. **Functional testing** (label – ‘Functional’)

It means that we need to verify the work of system, how the system behavior reflects the requirements and the understanding of user. Just ‘what the system does’. The following things should be tested:

- the buttons are working as expected (create, delete, search, disable, enable, next, etc.);
- the database is updated according to changes in portals;
- the system correctly responds the user’s actions (telemetry, actions, upgrades, roles, etc.);
- new features.

## 3. **Performance testing**

This kind of tests should be divided into two main parts: Performance of BE and Performance of FE.

### 3.1 *Performance of BE part* (label – ‘BE Performance’):

- many requests to the database at the unit of time (from 1 users as well as from many different users);
- the time of response measurement;
- the memory usage measurement;
- scale tests (when we increase the flows, and the system should increase performance proportionally);
- the data losses, data duplication during a big request flow;
- the data losses, data duplication during a big amount of data (a lot of telemetry);

### 3.2 *Performance of FE* (label - ‘FE Performance’):

Here the time of response from Portals can be tested. Make these measurements on a single separate server, where this data will be the most valid. Here we need to decide, what pages are the most important for such measurements

TO DO

## 4. **Recovery testing** (label – ‘Recovery’).

The aim of such tests is to be sure that the system will not lose the data in case of some unexpected events. There can be tested:

- disabling of DB (and check what will be with a system, is there some backup of database or not);
- disabling of the device, which is sending telemetry (check the losses);
- the connection loss between BE and FE.

## 5. **Security testing** (label – ‘Security’).

TO DO

## 6. **Compatibility testing** (label – ‘Compatibility’)

The following scenarios can be here:

- testing of different versions (iPhone 4, 5 etc..), types (iOS, Android, etc.) of devices;
- selene testing;
- testing in old browsers.

## 7. Usability testing (label – 'Usability')

There the following tests should be:

- labels of fields to be sure that they are understandable;
- the sizing of elements;
- the understandable helps;
- overlapping in case of big data;
- the color;
- the appearance of the same fields in different parts of system;
- the screen 'overload' by elements;
- convenient scrolling, etc.;
- good looking in mobile size;
- absence of big not used empty areas (for example, one button at the right side of completely empty line);
- intuitively understandable interface (for example, when newcomer can easily propose what the UI element means)

## 8. Installation testing (label – 'Installation')

The testing of arrow connect Application on mobile devices, how to install and help for it.

Moreover, the Smoke testing and Regression testing should be done.

Smoke testing is needed to test the work of main parts of system. If Smoke testing is failed, the product should not be released.

Regression testing – to be sure that old functionality wasn't broken by new deliveries and features.

We need to decide what tests should be run in the scope of Smoke and Regression testing and mark them as 'Regression' and/or 'Smoke'.

All (or the main part of) regression and smoke tests should be automated just to save time and test as much as possible to be sure that the product has good quality.

## Prioritization.

We should have a system of prioritization for tests for choosing what tests are the most important, what - less. To understand - what should be tested for the first time.

There should be Critical, Major and Minor prio, all tests should be marked by that prio in Jira.