User Acceptance Testing

User Acceptance Test or User Acceptance Testing (UAT) is a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This report is to illustrate the application of UAT in developing, testing, and releasing a new or updated R package.

User Acceptance Test for R package release

In the DTAP process (development → test → acceptance → production), which is part of a software development process, the release of a new version of an R package is at the final step where the deployment needs to meet certain requirements. The "acceptance" portion in this process comes from both *humans* (developer and clients) and *machines* (a list of rules that the software must conform). Of course, human testing involves machines as well, and this distinction is made for referring to a systematic way of testing.

Human testing

The development of a software component on a local machine has folders containing a set of functions in a structure like this

```
FOLDER-BUILD
FOLDER-PUB
Batch file to build
```

where the testing of the software occurs in the three parts.

A set of R functions are written in `FOLDER-BUILD` with a version control, and then the functions are tested manually in the R console or other environment. It is important at this stage that the test involves random data, different data sets, and the extreme cases that the input data may have. Once there is an acceptance from the developer, `FOLDER-PUB` hosts a copy of the functions to publish and then the building with the `batch file`.

The batch file has the instructions in a sequential list to build the package that typically are R core functions to load files, write or recreate an object to a file, and the creation of a skeleton for a new source package.

Once the package is built then comes the documentation of the functions with files in a Latex format that conform the manual. The machine testing, which comes afterwards, will check among other things whether there is a correspondence or not between the script codes and the documentation.

Machine testing

Rtools allows performing a machine testing of R packages within the MS Windows operating system. First the package is constructed as a tarball file with the command line and by typing `R CMD build`. Then the `R CMD check` performs different types of tests on this file based on pass/fail results where "fail" involves errors, warnings, and notes. The option `-as-cran` applies the most strict rules on the package and allows a successful submission for a publication on the CRAN repository whenever there is any fail.

An example of a successful testing is in the `.log` file given as appendix where the checking starts in the `DESCRIPTION` file with the basic information about the package. Then there is also a testing of things like whether the package can be loaded and unloaded, and consistencies both in the code and the documentation including meta data.

UAT for users in GitHub

As with machine testing, the acceptance tests among users of the R package should reveal as well a straightforward yes/no or pass/fail results. This means that the user should be able to download, install, uninstal, and run the software without any errors.

Another important component of the user acceptance is counting with a shared infrastructure that team members administer, use on a day-to-day basis and can reflect on and implement for others. GitHub is not only a git repository, but it is also a tool suitable for educational tasks such as periodical reflection and implementation for others.

GitHub storages the information and it can be used as well as backup for disaster recovery. In many cases GitHub is the place where the users report bugs, and where ideally users should become developers of the package as well.

UAT exercise

1. install the beta version of the package from GitHub plus dependencies into the R environment
2. load the package and run the scripts of the `README.md` file
3. run the program with different datasets
    - if an error occurs then consult the manual to fix the input data
    - if the data introduced conforms the requirements from the manual and the error remains then report the bug

A successful user acceptance test has no errors, and the user is willing to use the package in his/her own work.