QTO APPLICATION CONCEPTS

1. INTRO

1.1. Purpose

The purpose of this document is to present the concepts and the business logic of the qto application for this current version.

1.2. Audience

This document could be of interest for any potential and actual users of the application. Key-users must read, understand and even be able to present and explain the contents of this document.
Developers and Architects working on the application MUST read and understand this document at least to the extend of their own contribution for the application.

1.3. Related documentation

This document is part of the qto application documentation-set, which contains the following documents:
- readme_doc-0 - the initial landing readme doc for the project
- userstories_doc-0 - the collection of user-stories used to describe "what is desired"
- requirements_doc-0 - the structured collection of the requirements
- system_guide_doc-0 - architecture and System description
- devops_guide_doc-0 - a guide for the developers and devops operators
- installations_doc-0 - a guide for installation of the application
- enduser_guide_doc-0 - the guide for the usage of the UI (mainly) for the end-users
- concepts_doc-0 - the concepts doc

All the documents should be updated and redistributed in combination of the current version of the application and should be found under the doc/md directory.

2. BUSINESS LOGIC

2.1. Projects management

You can manage multiple projects with the qto tool. Each project has its own data directories, database storage and configurations. You could also have different environments named dev, tst, prd for each project separately.
As the tool is backwards compatible you could have different instances of the qto projects with different versions (and set of features) operating against different project (each one in its own version).
You must pre-set the configuration variables of an qto project each time you start working on a project from the shell.

2.2. Issues / Issue items / items

Issue item is the shortest possible description of task , activity , note or anything requiring distinguishable and preferably measurable action or producing verifiable outcome.
Issues could be different types - tasks, activities, notes etc.
Each issue MUST BE assigned to one and only one person.
Issues as any items can be interlinked by simply copying the name of the item concatenated with dash and it's id - for example the concepts_doc-12 points to the next section and the problems-1807031157 to a specific problem.

2.2.1. Categories

Each issue item could be categorised under one and only one category. One category might have 1 or more issues.
The categories could contain letters ,numbers, dashes.

2.2.2. Issues status

You could define whatever statuses you like. As qto has been built to develop itself (how sarcastic could that be ;o), that is for tracing the progress of a software project the following "default" statuses have been used:
01-eval - for evaluate the issue
02-todo - for decided to do the issue
03-wip - for the issue is being in work in progress mode
04-diss - for discard the issue
05-tst - for the issue is being in testing like mode
06-onhold - for the issue is being kept onhold for the time period
07-qas - for the issue is being quality assured
09-done - for the issue is being done / completed

2.2.3. Issues management via time intervals

The issues are basically organised into the following time intervals:
- release_issues - the issues to be handled till the next release (usually, but not always a 2 weeks period)
- monthly_issues - the issues to be handled during the next month
- yearly - the issues to be handled during the next year
- quinquennially
- decadally

So that in the end of each previous time period you could go trough the issues of that period and transfer up and down in the time scale.
The product instance owners have an automated way of transferring the issues from one table to another described in the maintenance_guide_doc-200111060442 document.

2.3. Questions

Sometimes during the workings of your project you encounter problems, which are complex enough not to allows the definition of an issue. In those cases it would be more rational to just register the question, discuss it or review it later on, and define the issue as soon as the problem domain is understood and even possible issue solution could be proposed.

2.4. Ideas

Your organisation might collect, sort and evaluate ideas so that they could be later one used as the row data for issues (aka concrete work descriptions to be followed).

2.5. Problems

Quite often the row input material for the issues are the problems encountered - the better you collect, describe and prioritise the list of problems to tackle the better you will be able to organise the issues to be completed. You can interlink problems with the description fields of any other item, for example: problems-1807031157

3. DEFINITIONS

This section contains definitions of terms within the context of the qto application.

3.1. Release

A qto release is the artefact you can download from the following GitHub page:
https://github.com/YordanGeorgiev/qto/releases.