Table of Contents

ISSUE-TRACKER CONCEPTS

1. INTRO

1.1. Purpose

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

1.2. Audience

This document could be of interest for any potential and actual users of the application. Keyusers 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 Issue-Tracker application documentation-set, which contains the following documents:
- UserStories - the collection of userstories used to describe "what is desired"
- SystemGuide - architecture and System description
- DevOps Guide - a guide for the developers and devops operators
- Installation Guide - a guide for installation of the application
- Features and Functionalities - description of the current features and functionalities
All the documents should be updated and redistributed in combination of the current version of the application and should be found under the following directories:
doc/md
doc/pdf
doc/xls
according to the file format used for the documentation storage.

2. CORE CONCEPTS

2.1. Time intervals based time-management

The time rules our lifes. Issue-tracker is aimed to help you and your organisation to manage your time and resources in the best possible way YOU CAN. You must make sure that you clearly define and state your issues and update their information based on the events from the real world.

2.2. Personal responsibility

You should assign 0 or 1 persons responsible for an issue at a time. Of course if you keep tool long an issue to be assigned to nobody it simply means that you are not fully committed to its completion.

3. BUSINESS LOGIC

3.1. Projects management

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

3.2. Increase the date for all projects

to increase the date for all the projects at once use the following oneliner.

3.3. Issues / Issue items / items

Issue item is the shortest possible description of task , activity , note or anything requiring distinguishable and prerferable measurable action or producing verfifiable outcome.
Issues could be of different types - tasks, activities, notes etc.

3.3.1. Categories

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

3.3.2. Issues status

You could define whatever statuses you like. As issue-tracker 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.

3.3.3. Issues management via time intervals

The issues are basically organised into the following periods:
- daily
- weekly
- monthly
- yearly
- 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.

3.4. Problems

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 problem, 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

3.5. Questions

Sometimes during the workings of your project you might be simply asking questions, 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(s) addressing this question later on.