The Project Web Concept

System Design presented as

Documents containing Specification Cards

See sample at http://greiterweb.de/spw/xs_WissDB/1/1.htm

The idea is that each Project should have a Project Web

where ALL and ONLY current document versions are presented

– indexed and heavily hyperlinked –

in HTML.

Gebhard Greiter, 2011

Notation and Terminology

The word **_Concept_** – when used in this document – always refers to a typed concept in the following sense:

Each concept specified in a software design paper is given a name starting with a prefix telling you the type of the concept. Prefix semantics for the concepts currently supported by C_projweb are:

D_(…)	=	A logical name of a document (a so-called Semi Title, need not be unique)
R_	=	A requirement
C_	=	A software component
S_	=	A service offered by software component (and callable via <u>at least</u> an API)
$_	=	An installation parameter
m_	=	A task to be performed manually
P_	=	A business process or a technical process supporting a business process
A_	=	An actor (in the sense of a user role or a system's role)
V_	=	A view on data. It should be definable via SQL (so that standard reporting tools can be applied).
B_	=	A business object type: Given any specific **B_x**, all instances of type **B_x** are owned by a unique Component. Creating instances is possible via this component only.
D_	=	A data type
E_	=	An Entity Type (a relational table)

A **_typed name_** is an identifier following one of the prefixes defined in this list.

The prefix, denoting the type of Concept identified, may or may not be followed by a blank. The blank, if there, will tell the tool that this specific occurrence of the concept name should not become a hot spot because here the concept is defined (whereas everywhere else it is referred to).

Hyperlinks created by the tool C_projweb.exe always start

- at an occurrence of a typed name,
- or at a navigational element added by the tool itself in order to provide useful hot spots to the reader of the web page.

To use type concept prefixes is all the author of a design paper needs to do in order to allow the tool to create a presentation in which human readers can navigate to the spec via one mouse click only.

If, on the web page, a typed name is not a hotspot, the corresponding concept is not yet specified

(which also is useful information). The tool will mark such occurrences by adding **[?]**. This will help the Quality Manager to see how incomplete the current design may be.

Note also: What we see as a *component C_...* can be defined to some extent in your project. The general rule however is: Everything not yet typed in the table above should be classified as a component, e.g. interfaces with a non-trivial protocol.

Suggestion for a Structure of your Project Folder

All data in the **Project Folder** should be kept under a deepest node of the following structure:

- **Discussion**
    - One File per Topic

- **Management**
    - Schedule
    - Budget
    - QM Reports

- **Requirements**
    - Agreed
    - Emerging

- **Design**
    - Agreed
    - Emerging

- **Code**

- **Test**
    - Design
    - Drivers
    - Data
    - Latest Test Result

- **Project Web** (ALL documents shown in their currently relevant version)

- **YY_MM_DD_ VersionName** (none, one, or more such **F R O Z E N** Product Version)
    - Requirements Agreed
    - Design
    - Code
    - Test
    - Release Notes

- **Products in Use**
    - OpenSource
    - Licensed

- **Products under Evaluation**
- OpenSource
- Licensed

- **Outdated**

Each file or subtree that is meant to be frozen (because it was **published to a person different from the author**) has to have a name of the form

YY_MM_DD_ TOPIC

such that TOPIC is content characterization and YY_MM_DD_ is the date of publication.

Folders not frozen may contain a subfolder **Outdated**. <u>Files or folders not worth to be read any more should either be deleted or should to go there:</u> **Please, do not keep them elsewhere**.

The **Discussion** folder is a set of subfolders "On TOPIC" which contain files (mails and meeting notes) with an identifier of the form:

YY_MM_DD_ TOPIC

such that TOPIC is chosen by you. Files containing meeting notes should be named

YY_MM_DD_ Meeting Notes

so that they can be detected automatically.

Each **Requirements** folder only contains files named

Requirements on TOPIC

If the file is to be read only, the name is to start with a date (seen also as a version number):

YY_MM_DD_ Requirements on TOPIC

Each **Design** folder should contain files named

Design of TOPIC

If the file is to be read only, the name is to start with a date (which can be seen as a version number):

YY_MM_DD_ Design of TOPIC

Each folder may contain subfolders with names prefixed by **x_** :

x_Actor

The prefix x_ is to say: Ignore me - this is temporary stuff not interesting to any other person than the one that created this folder (a kind of scratch area). We will not loose anything by deleting these data as soon as the Actor is no longer available.

The purpose of such a folder is to have a place where we can hide away everything that only would confuse persons different from the author.

PLEASE NOTE: The latest version of a document – even it is work in progress far from being complete or perfect in any sense – must NEVER be stored in such a place.

The **Project Web** folder should be maintained and controlled by the project's Quality Manager.

One of his responsibilities should be to ensure that at any time ALL relevant documentation can be found there – and ONLY in the unique version which is the currently relevant one.

No longer can then any team member work by mistake with a wrong or outdated version of any such document or not know with documents actually exist.

A sample Project Web can be found at

http://greiterweb.de/spw/xs_WissDB/1/1.htm