

Requirements Definition Tabelle (Vorschlag)

Column name	Type	Nulls
rd_id	serial	no
rd_projid	char(8)	no
rd_refid	char(8)	yes
rd_text	varchar(255)	yes
rd_class	char(1)	yes
rd_prio	smallint	yes
rd_weight	smallint	yes
rd_prefid	char(8)	yes
rd_minval	char(32)	yes
rd_maxval	char(32)	yes
rd_valid	char(32)	yes
rd_comment	varchar(255)	yes
rd_create	datetime year to minute	yes
rd_modified	datetime year to minute	yes
rd_author	char(8)	yes
rd_module	char(8)	yes
rd_source	char(32)	yes
rd_outdate	date	yes

Erläuterungen:

- rd_id ist ein eindeutiger Schlüssel (serial 1...32768 vermutlich)
- rd_projid ist die ID des Projektes (siehe p_name in projekte)
- rd_refid ist die vom Autor vergebene Referenz-Kennung der Anforderung (eindeutig für das Projekt)
- rd_text ist der Anforderungs-Text (wohlgeformt nach IEEE/VDE)
- rd_class ist die "Klasse": (F)unktionale Anf., (L)eistung, (Z)uverlässigkeit, (Q)ualität (man kann sich hier gerne noch mehr überlegen)
- rd_prio ist "MUSS" "KANN" "NICE" oder meinetwegen auch 5, 2, 1
- rd_weight ist ein Gewicht von 0 = unwichtig bis 10 = superwichtig
- rd_prefid ist die "parent refid", d.h. die rd_refid des darüberliegenden Knotens einer eventuellen Hierarchie
- rd_minval ist ein Mindest-Wert mit Einheit der geforderten Eigenschaft, falls diese irgendwie meßbar ist (z.B. Antwortzeit "0 s")
- rd_maxval ist ein entsprechender maximal-Wert (z.B. Antwortzeit "3 s")
- rd_valid sagt, ob die Anforderung gültig ist und wer das gesagt hat (freier Text)
- rd_comment ist ein Kommentar
- rd_create ist der Zeitpunkt der Erstellung
- rd_modified ist der Zeitpunkt der letzten Änderung
- rd_author ist der persons.ident Wert des letzten Änderers
- rd_module ist der Systembereich, für den die Anwendung gilt, z.B. "WE" wie Wareneingang.
- rd_source ist eine Quellenangabe, z.B. ph 4.3.5 für Pflichtenheft Kap. 4.3.5.
- rd_outdate ist das Datum, wann diese alte Version der Anforderung ungültig wurde, weil es Änderungen gegeben hat. Wichtig: Dieses Feld enthält bei der aktuellen Version <NULL>!

Konzepte

Schon bei kleineren Projekten kommen schnell 50...100 Anforderungen zusammen, und um den Überblick zu behalten, braucht man eine Struktur und Werkzeuge.

Die Kerndaten sind natürlich rd_text, rd_prio, rd_weight, rd_minval und rd_maxval. Diese definieren, was gefordert wird, ob es "lebensnotwendig" ist (rd_prio) und wie wichtig die Anforderung ist (rd_weight). Messbare Parameter sind rd_minval und rd_maxval.

Für die Strukturierung und Verwaltung sorgen natürlich die Projekt-ID (rd_projid) sowie der Identifier rd_refid. Weiterhin wird die Anforderung klassifiziert und sortiert durch rd_prefid (schafft eine Hierarchie) sowie rd_class und rd_module (sorgen für die fachliche Einordnung).

Wichtig für die Validierung und Verifizierung sind rd_valid, rd_author und rd_source. Ohne diese würde man immer wieder rätseln, wo eine Anforderung eigentlich herkommt. Alte Versionen können aufbewahrt werden, dann ist das Feld rd_outdate zu setzen.

Im Moment werden die Inhalte der Felder zunächst intuitiv verwendet. Wenn mehr Erfahrungen vorliegen, können feste Konventionen definiert werden, z.B. zum Feld rd_source. Auch die Hierarchie-Kennzeichnung mit rd_refid und rd_prefid (parent refid) ist nicht gerade genial, aber erstmal geht es so.

Verbesserungs-Vorschläge

- Automatische Vergabe von rd_refid, einfaches Handling von rd_prefid.

Brunthaler
November 2008