

enso
managers

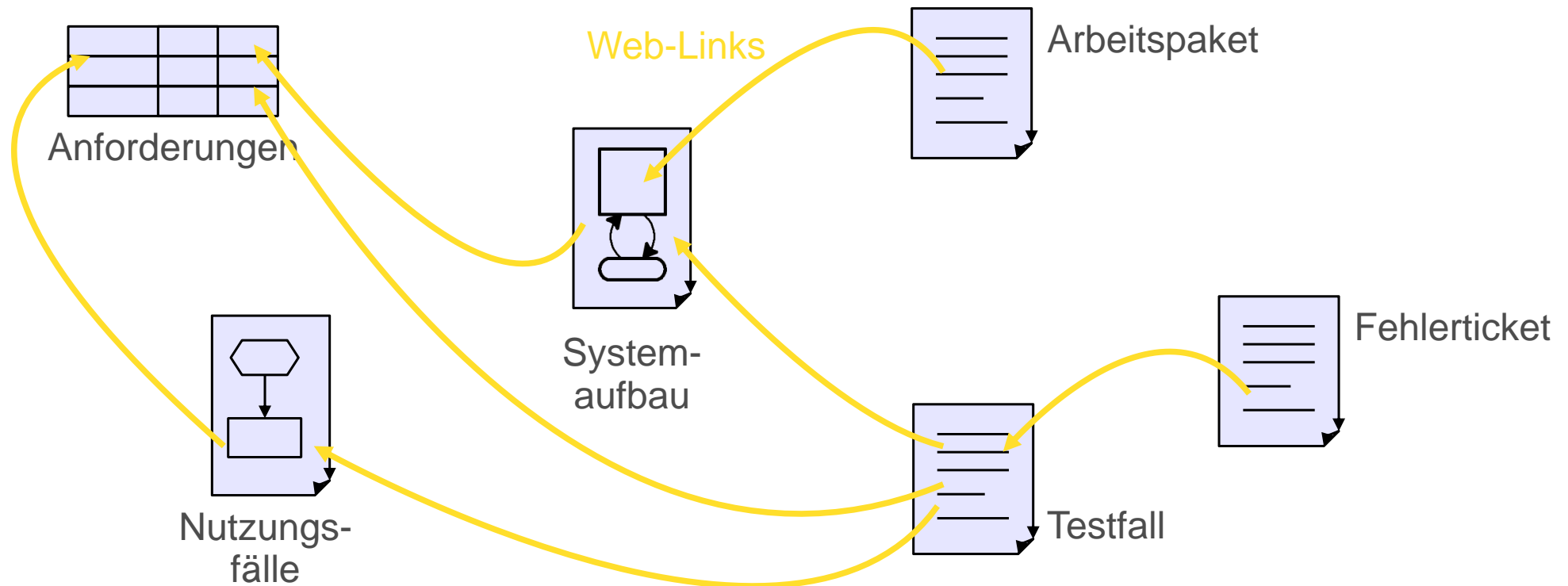
Nachvollziehbarkeit sichern: Testfälle auf Anforderungen beziehen

Aufgabenstellung

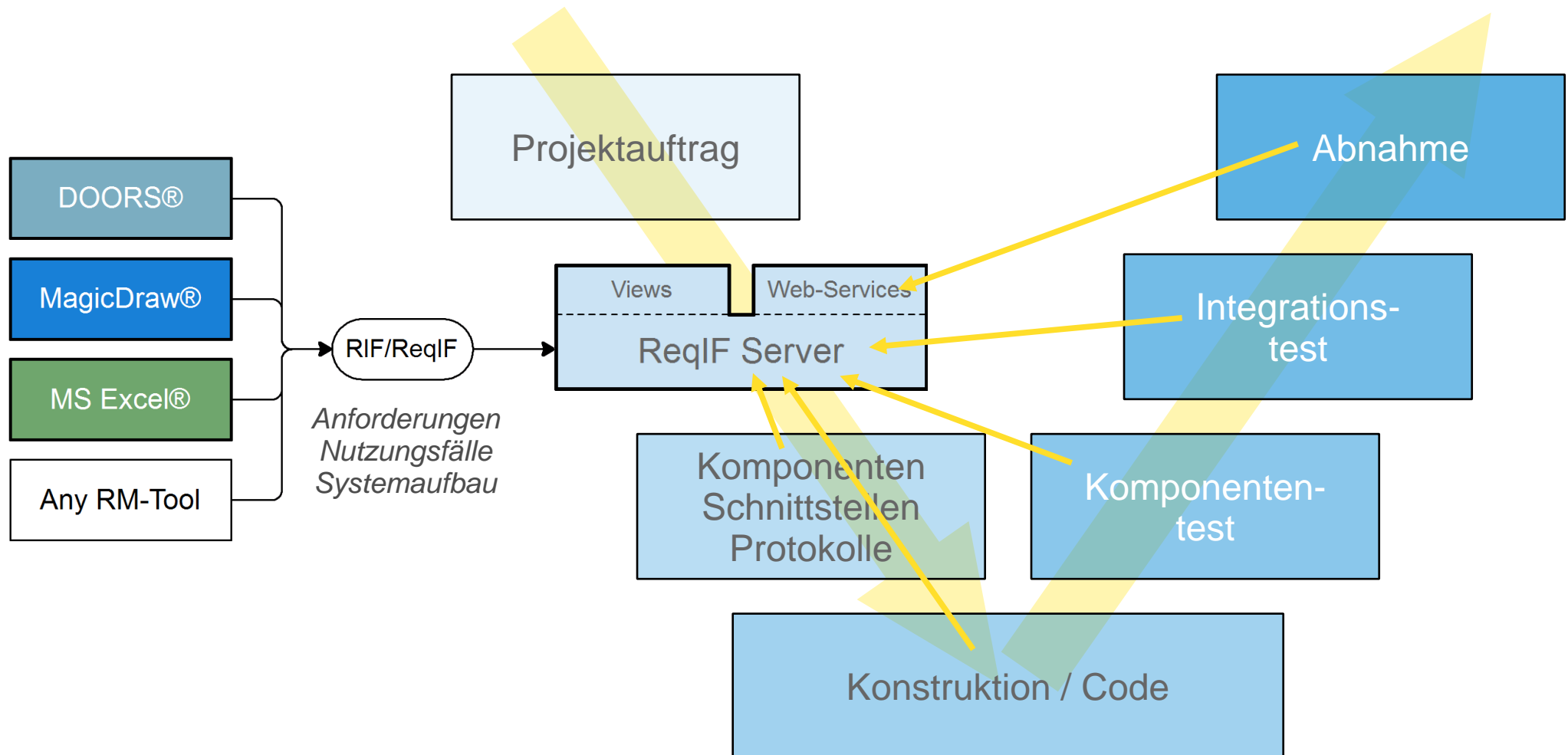
- Testfälle auf relevante Anforderungen, Use-Cases, GUI Entwürfe und andere Elemente des Lastenhefts oder des Systementwurfs beziehen – auch wenn das eingesetzte Anforderungsmanagement- oder Modellierungs-Werkzeug das nicht unterstützt.
- Anforderungen und andere Elemente des Lastenheftes historisieren, damit zu jedem Zeitpunkt nachvollziehbar ist, welche Revision für welches Software-Release gültig ist bzw. war.
- Verweise sollen revisionsgenaue und permanente Web-Links sein – für eine einfache Werkzeugintegration ohne Kopieren von Inhalten.
- Bei Änderung einer Anforderung oder des Systementwurfs („Change Request“) den Qualitäts-Verantwortlichen automatisch benachrichtigen, damit abhängige Testfälle und Testplanung angepasst werden können.
- Testfall-Struktur aus Lastenheft-Struktur erzeugen

Nachvollziehbarkeit im Entwicklungsprozess erreichen

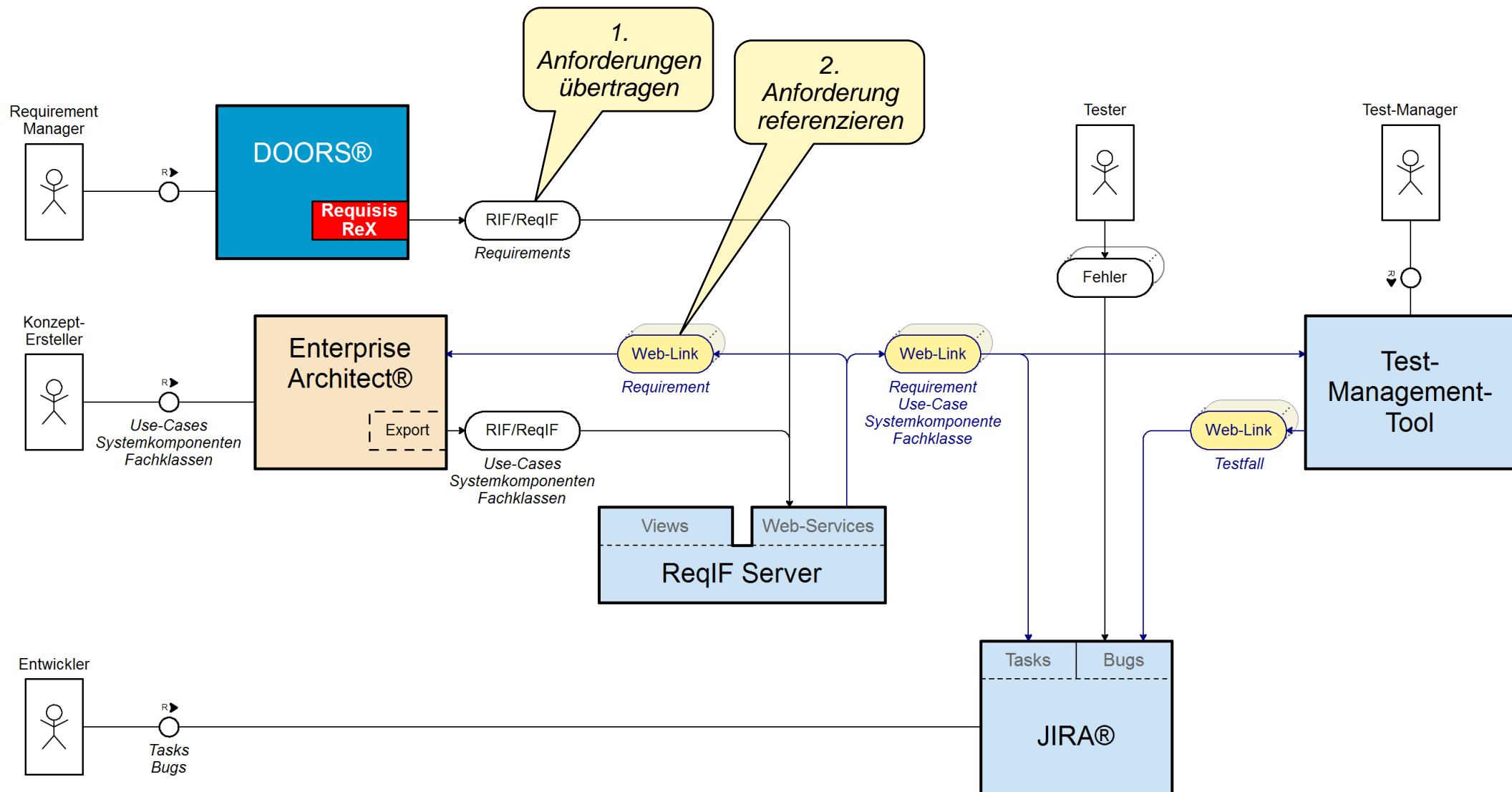
- In nachfolgenden Entwicklungsschritten auf bestehende Arbeitsergebnisse in Anforderungsmanagement- und Modellierungs-Tools verweisen.
- Arbeitsergebnisse historisieren und Verweise revisionsgenau anlegen.



Konzept: Spezifikation im Netz publizieren und Bezüge herstellen



Mit eingeführten Werkzeugen einen durchgängigen Informationsfluss schaffen (Beispiel)



Rolle des ReqIF Servers

- Der ReqIF Server übernimmt die Inhalte von Lastenheften und Systementwürfen mittels des internationalen OMG Standards „Requirements Interchange Format“ (RIF/ReqIF), der von der deutschen Automobilindustrie initiiert wurde.
- Die ReqIF Daten enthalten Struktur und Inhalt, so dass keine weitere Vorbereitungen im ReqIF Server anfallen.
- Der ReqIF Server stellt pro „Specification Object“ (Anforderung, Nutzungsfall, ...) und pro zugehöriger Liste eine Web-Adresse bereit, über die Inhalte im HTML-, XML- oder JSON-Format angefordert werden können.
- Es kann jeweils die neueste Revision (rev=0) oder eine bestimmte Revision adressiert werden (rev>0).
- Bei wiederholtem Import wird nur dann eine neue Revision eines Objekts angelegt, wenn es sich unterscheidet. So kann eine historisierte Datenhaltung aufgebaut werden, selbst wenn das liefernde Werkzeug nicht historisiert.
- Mit dem ReqIF Server kann Nachvollziehbarkeit erreicht werden – nach dem neuesten Stand der Wissenschaft und Technik.
- Nähere Informationen unter www.reqif.de.

Web-Links liefern Listen und Objekte – hier im HTML Format

Detailansicht einer Anforderung

[↳ Auto-import from 'hot-folder'](#) | [details](#) |
 If a RIF/ReqIF file or ZIP/REQIFZ file container is placed in a certain hot-folder, it is automatically imported. The result code is placed in a file with the same name .. or a file extension is added (filename.reqifz --> filename.reqifz.rc-0).

[↳ Server-side filtering - sorting - paging](#) | [details](#) |
 Based on user-supplied criteria, data is selected and delivered.

- 'Filtering' includes searching and selects a sub-set of the data to be delivered. Data can be filtered by lines (object instances) and/or columns (attributes).
- 'Sorting' sequences the data to be delivered (alpha-up, alpha-down, newest-first etc).
- 'Paging' delivers data in slices of maximal size (first, next, previous, last or directly addressed).

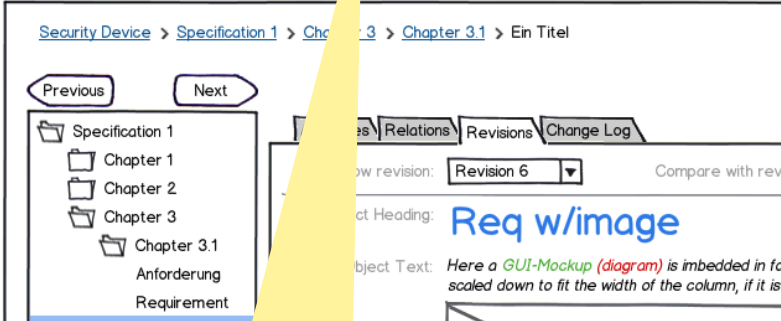
These operations are implemented on server side to support large data volumes as found in enterprise environments.

[↳ Highlight changes](#) | [details](#) |
 Highlight changes (additions and omissions) between revisions – both in table and detail views.

Benefits:

- Easily compare different revisions of requirements similarly to word processors.

Diagram:



Anforderungsliste

Server-side filtering - sorting - paging

ReqIFForeignID: REQ_147

ReqIFName: [↳ Server-side filtering - sorting - paging](#)

ReqIFText: Based on user-supplied criteria, data is selected and delivered.

- 'Filtering' includes searching and selects a sub-set of the data to be delivered. Data can be filtered by lines (object instances) and/or columns (attributes).
- 'Sorting' sequences the data to be delivered (alpha-up, alpha-down, newest-first etc).
- 'Paging' delivers data in slices of maximal size (first, next, previous, last or directly addressed).

These operations are implemented on server side to support large data volumes as found in enterprise environments.

Status: 30_submitted

Priority: 2_medium

Delivery:

Requestor: michael.holzner@icondu.de, od@enso-managers.com

Fit Criterion:

Ticket No:

Effort Server [h]: 0

Effort iLAH [h]: 0

Reviewer Status:

Reviewer
Comment:

Author Status:

Author Comment:

Web-Links liefern Listen und Objekte – hier im JSON Format

```

{"type": "SPEC-OBJECT",
 "specType": "Requirement",
 "values": [{"type": "ATTRIBUTE-VALUE-SIMPLE",
  "attributeDefinitionSimple": "req-object-ID",
  "theValue": "RBR_006",
  "description": null,
  "identifier": "RBR_006-object-ID",
  "lastChange": "2011-10-02T18:21:00+02:00",
  "longName": "Object ID",
  "createdAt": "2011-10-08T11:20:40+02:00",
  "createdBy": null,
  "lastChangeBy": null,
  "url": "projects/ACP-aa3027020000e9a8012f0d87d7ad2719-16/specObjects/RBR_006/attributes/RBR_006-object-ID",
  "revision": 1,
  "deleted": null},
 {"type": "ATTRIBUTE-VALUE-EMBEDDED-DOCUMENT",
  "attributeDefinitionComplex": "req-object-heading",
  "xhtmlContentObject": "<strong>show specObject details</strong>",
  "description": null,
  "identifier": "RBR_006-object-heading",
  "lastChange": "2011-10-02T18:21:00+02:00",
  "longName": "Object Heading",
  "createdAt": "2011-10-08T11:20:40+02:00",
  "createdBy": null,
  "lastChangeBy": null,
  "url": "projects/ACP-aa3027020000e9a8012f0d87d7ad2719-16/specObjects/RBR_006/attributes/RBR_006-object-heading",
  "revision": 1,
  "deleted": null},
 {"type": "ATTRIBUTE-VALUE-SIMPLE",
  "attributeDefinitionSimple": "req-object-text",
  "theValue": "\nshow a specObject in a form with its details:\n- attributes\n- relationships\n- hierarchies\n\nTo be discussed:\n- show in a tab, a popup or in a separate frame. \t\t\t\t",
  "description": null,
  "identifier": "RBR_006-object-text",
  "lastChange": "2011-10-02T18:21:00+02:00",
  "longName": "Object Text",
  "createdAt": "2011-10-08T11:20:40+02:00",
  "createdBy": null,
  "lastChangeBy": null,
  "url": "projects/ACP-aa3027020000e9a8012f0d87d7ad2719-16/specObjects/RBR_006/attributes/RBR_006-object-text",
  "revision": 1,
  "deleted": null},
 {"type": "ATTRIBUTE-VALUE-ENUMERATION",
  "attributeDefinitionEnumeration": "req-status-",

```


Argumente

- Projektarbeit erleichtern durch gezielten Zugriff auf relevante Informationen – statt Lastenheft als PDF auf einem File-Server.
- Nachvollziehbarkeit erreichen durch Verweise auf individuelle Anforderungen und andere Elemente des Lastenhefts oder Systementwurfs – revisionsgenau.
- Informationsfluss verbessern und Fehler vermeiden durch automatische Benachrichtigung bei Änderungen.
- Durchgängigen Informationsfluss mit Zugriff auf einzelne Datenelemente aufbauen – ohne eingeführte Werkzeuge abzulösen.
- Grundstein einer service-orientierten Architektur für die Werkzeugkette legen.
- Show-Case vorstellen und Wettbewerbsvorteile erzielen.



Haben Sie Fragen?

Kontakt

Dr.-Ing. Oskar v. Dungern
+49 173 670 9958
od@enso-managers.com

Anschrift

enso managers GmbH
Charlottenstrasse 68
10117 Berlin
Deutschland

Information

www.enso-managers.com
www.enso-factory.com
www.reqif.de



Ist das interessant für Sie?