

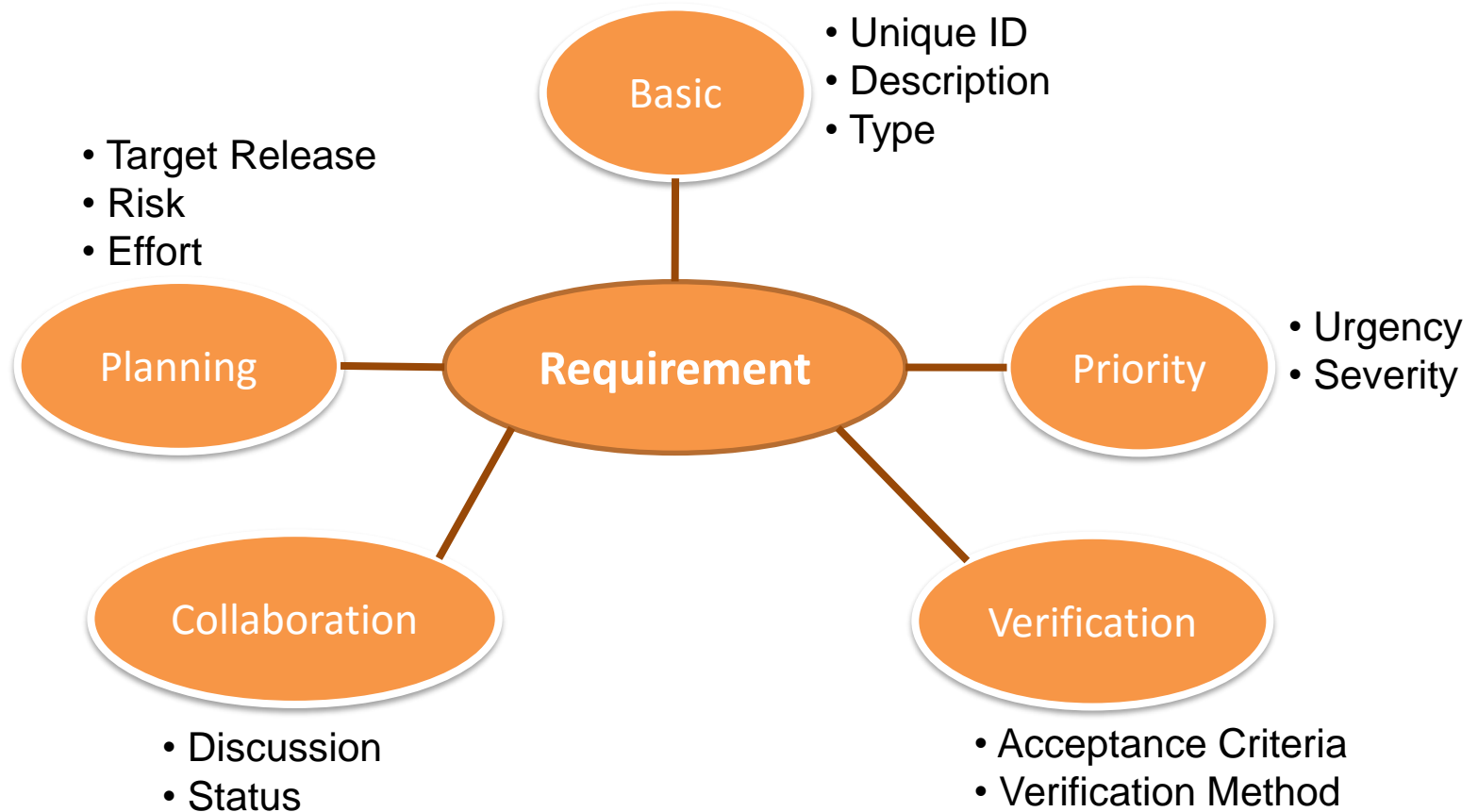
# ReqView

## Custom Requirement Attributes

[reqview@eccam.com](mailto:reqview@eccam.com)



# Requirement Attributes



# Elementary Attribute Types

- *int* - integer number
- *bool* - logical value, *true* or *false*
- *string* - string value
- *date* - date stored as [ISO 8601](#) string
- *real* - floating point number
- *enum* - enumeration value
- *xhtml* - [XHTML](#) string



# Predefined Requirement Attributes

- *id* : requirement identifier (*string*)

- Unique in the document
- Read only and automatically incremented

🔍 ID
64

- *heading* : section name (*string*)

- Should not be filled for requirements, only for sections (requirements grouped to features)

☰ Description
3.1.1.4 Encryption


- *text* : rich text requirement description (*xhtml*)

☰ Description
User <b>shall</b> be able to create a new empty file.




# Predefined Requirement Attributes


- *deleted* : delete flag  
(*bool*)

ID	Description
254 	User <del>shall</del> be able to create a new empty file.

- If a requirement is deleted then this flag is set to true. Otherwise the property it is not set.

- *discussion* : list of comments  
(*object*)

 Discussion

 Libor Bus, Eccam

This is not feasible, reject it.

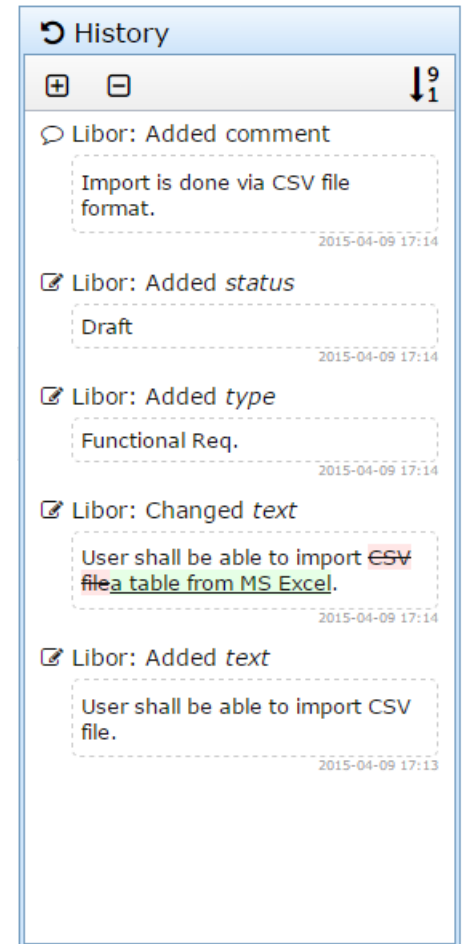
2014-03-28 17:04

- Each comment stores author's *email, name, and company*, comment *date* and *text*
- Last user's comment can be modified if is not followed by a comment made by another user.



# Predefined Requirement Attributes

- *history* : list of all requirement modifications (*object*)
  - Each modification stores name of the changed attribute, its old value, modification *date*, author's *email*, *name*, and *company*
  - History can be cleared to track changes since some project version



The screenshot shows a 'History' panel with a search icon, a close button, and a list of 9 items. The items are:

- Libor: Added comment  
Import is done via CSV file format.  
2015-04-09 17:14
- Libor: Added *status*  
Draft  
2015-04-09 17:14
- Libor: Added *type*  
Functional Req.  
2015-04-09 17:14
- Libor: Changed *text*  
User shall be able to import ~~CSV file~~ a table from MS Excel.  
2015-04-09 17:14
- Libor: Added *text*  
User shall be able to import CSV file.  
2015-04-09 17:13



# Definition of Elementary Attributes

```
<attributeName>: {  
  type: <string>,  
  name: <string>,           // optional  
  readOnly: <bool>         // optional  
},
```

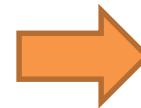
- *attributeName* is a chosen unique identifier of the custom attribute, e.g. “status”
- *type* describes attribute type, e.g. “int”, “date”, ...
- *readOnly* is an optional flag set to *true* if user should be prevented from changing the attribute value



# Example: Effort Estimation

- Define an attribute for effort estimation of each attribute.
- Enables computing the effort estimation per a feature (group of requirements) and total effort estimation of the whole product

```
"effort": {  
  "name": "Effort [MD]",  
  "readOnly": false,  
  "type": "int"  
},
```



Effort [MD]
20

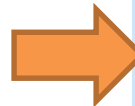




# Example: Acceptance Criteria

- Define an attribute for rich text description of the requirement acceptance criteria
- Allows to discuss and approve stakeholder requirements together with acceptance tests.

```
"acceptance": {  
  "name": "Acceptance",  
  "type": "xhtml"  
},
```



Acceptance

Rich text editor toolbar with icons for undo, redo, link, unlink, list, and text formatting (bold, italic, underline, strikethrough).

This is a free text description of the **verification method** for the requirement:

1. First do this
2. Then do that
3. What to finally check



# Example: Assign Requirements to Milestones

- Define a boolean flag for each milestone in the project.

```
"Rel-1.0": {  
  "name": "Release 1.0",  
  "type": "bool"  
},
```

- Set this flag to true if a requirement or a feature shall be delivered in that milestone.
- Enables filtering requirements assigned to a given project milestone.



# Definition of Enumeration Attributes

```
<attributeName>: {  
  type: "enum",  
  name: <string>,           // optional  
  values: [  
    { key: <string>, label: <string>, default: <bool> }, ...  
  ],  
  readOnly: <bool>,        // optional  
  multiValued: <bool>     // optional  
}
```

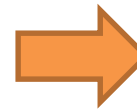
- *multiValued* is an optional flag set to *true* if the attribute can have multiple enumeration values.
- *values* is an array of enumerated values. Each value shall have a unique string *key* and can have optionally a text *label*, and an optional flag *default* denoting the default value.



# Example: Requirement State

- Define set of allowed requirement states to manage requirement lifetime

```
"state": {  
  "name": "State",  
  "type": "enum",  
  "values": [  
    { "key": "Draft" },  
    { "key": "Accepted" },  
    { "key": "Rejected" },  
    { "key": "Implemented" },  
    { "key": "Verified" }  
  ]  
}
```



State

Draft

**Draft**

Accepted

Rejected

Implemented

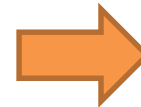
Verified



# Example: Allocation to Components

- Define set system components and assign requirements to one or more components

```
"components": {  
  "name": "Components",  
  "type": "enum",  
  "values": [  
    { "key": "ReqView Desktop" },  
    { "key": "ReqView Mobile" },  
    { "key": "Synchronizer" },  
    { "key": "Server" },  
    { "key": "Support" }  
  ],  
  "multiValued": true,  
},
```



Components

<input checked="" type="checkbox"/>	ReqView Desktop
<input type="checkbox"/>	ReqView Mobile
<input checked="" type="checkbox"/>	ReqView Synchronizer
<input type="checkbox"/>	ReqView Server
<input type="checkbox"/>	ReqView Support Site



# Document Attributes

Project Help

- New Project...
- Close Project
- Add Document...
- Delete Document...
- Document Properties...
- Document Attributes...**
- Project Traceability...

**Document Attributes**

In this dialog you can customize attributes for the current document.

```
{
  "status": {
    "name": "Status",
    "readOnly": false,
    "type": "enum",
    "values": [
      {
        "key": "N/A"
      },
      {
        "key": "Draft"
      },
      {
        "key": "Accepted"
      },
      {
        "key": "In Progress"
      },
      {
        "key": "Rejected"
      },
      {
        "key": "Implemented"
      }
    ]
  }
}
```

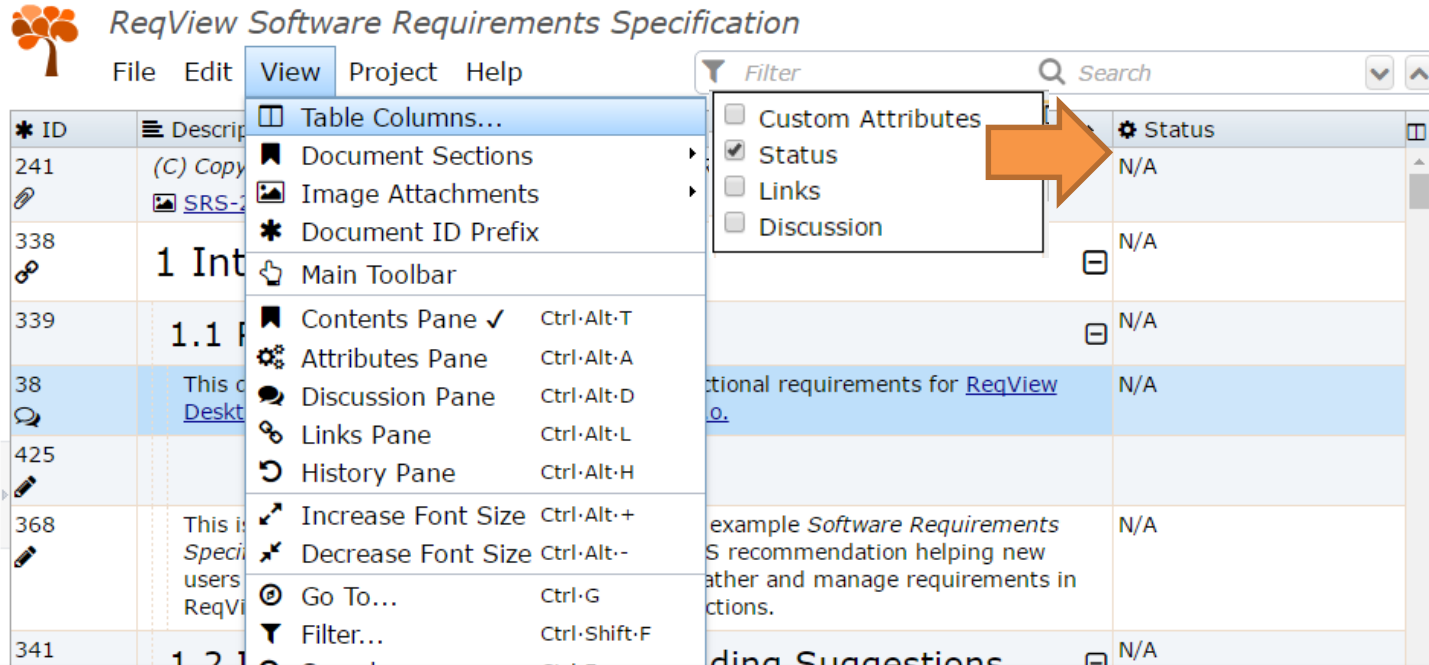
Edit attributes definition in JSON text format.

Attributes definition is validated before changing.

Check OK Cancel



# Show or Hide Attribute Columns



The screenshot shows the ReqView Software Requirements Specification application. The 'View' menu is open, displaying options for 'Table Columns...', 'Document Sections', 'Image Attachments', 'Document ID Prefix', 'Main Toolbar', 'Contents Pane', 'Attributes Pane', 'Discussion Pane', 'Links Pane', 'History Pane', 'Increase Font Size', 'Decrease Font Size', 'Go To...', and 'Filter...'. The 'Table Columns...' option is selected, and a sub-menu is visible with checkboxes for 'Custom Attributes', 'Status', 'Links', and 'Discussion'. An orange arrow points from the 'Status' checkbox in the sub-menu to the 'Status' column in the table below. The table has columns for ID, Description, and Status. The 'Status' column contains 'N/A' for all rows.

ID	Description	Status
241	(C) Copy SRS-2	N/A
338	1 Int	N/A
339	1.1	N/A
38	This o Desk	N/A
425		N/A
368	This is Speci users ReqVi	N/A
341	1.2	N/A

*Custom Attributes* column summarizes for each requirement values of all defined attributes.



# Customize Attributes View

Read only attribute

Custom attribute columns

ID	Description	Custom Attributes	Is Req.	Status	Priority	Components	Deadline	Effort [MD]
28	<b>3 Features</b> This section describes the planned features of ReqView version 1.0.		<input type="checkbox"/>					
32	<b>3.1 ReqView Desktop</b> This section describes the features of ReqView Desktop component.	> Components: • ReqView Desktop	<input type="checkbox"/>			• ReqView Desktop		
55	<b>3.1.1 File Operations</b> This section describes ReqView Desktop file operations.	> Components: • ReqView Desktop > Status: Implemented	<input type="checkbox"/>	Implemented		• ReqView Desktop		
253	<b>3.1.1.1 Create New File</b>	> Components: • ReqView Desktop > Status: Implemented	<input type="checkbox"/>	Implemented		• ReqView Desktop		
254	User shall be able to create a new empty file.	> Components: • ReqView Desktop > Effort [MD]: 5 > Is Req.: <input checked="" type="checkbox"/> > Priority: High > Status: Implemented	<input checked="" type="checkbox"/>	Implemented	High	• ReqView Desktop		
255	User shall be able to save the opened document containing unsaved changes when creating a new file.	> Components: • ReqView Desktop > Is Req.: <input checked="" type="checkbox"/> > Status: Implemented	<input checked="" type="checkbox"/>					

Custom attribute value editor

Reorder or resize attribute columns by drag and drop.





# Change Values of Custom Attributes

Custom Attributes		Status
> Components:	• ReqView Desktop	Implemented
> Status:	Implemented	
> Components:	• ReqView Desktop	Optional
> Status:	Optional	Optional
> Components:	• ReqView Desktop	Optional
> Is Req.:	<input checked="" type="checkbox"/>	Unknown
> Status:	Optional	Urgent
> Components:	• ReqView Desktop	Optional
> Is Req.:	<input checked="" type="checkbox"/>	<b>Optional</b>
> Status:	Optional	Not Needed
> Components:	• ReqView Desktop	Optional
> Status:	Optional	Implemented
> Components:	• ReqView Desktop	Optional
> Is Req.:	<input checked="" type="checkbox"/>	
> Status:	Optional	
> Components:	• ReqView Desktop	Implemented
> Is Req.:	<input checked="" type="checkbox"/>	
> Status:	Implemented	

Change attribute values of all selected requirements at once.



# ReqView References

- ReqView Homepage  
<https://www.reqview.com>
- ReqView Desktop Online  
<https://www.reqview.com/apps/desktop/ReqViewDesktop.html>
- ReqView Desktop at Chrome Store  
<https://chrome.google.com/webstore/detail/reqview-desktop/cmbnhmeabefpflidnahpbghcejkbli>
- Support Site - online help, knowledge base and discussion forum  
<http://support.reqview.com/solution/categories>

