

# Reuse Techniques Jan Jancar





www.softacus.com info@softacus.com

### **WHY REUSE?**

- Many custom variants are required by the market
- **Cost savings**
- Consistent high quality
- Improve Time to market
- Revalidation & Recertification
- Consistency
- Scalability
- **Customer satisfaction**
- Ability to attract different customers



### Simple Copy & Paste

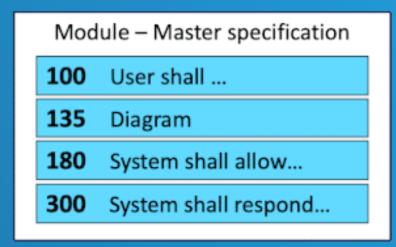
Simple technique for using existing data for a new project or product. Good option if reference to the original artifact is not needed.

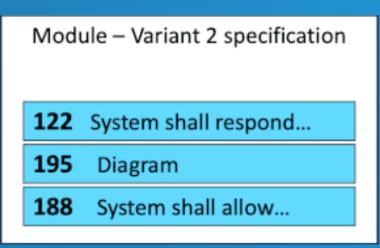
- Copied artifact will get a **new ID**.
- Ability to copy the links of the copied artifact.
- Ability to copy artifacts from specifications as well as other components and projects

#### **DISADVANTAGES**

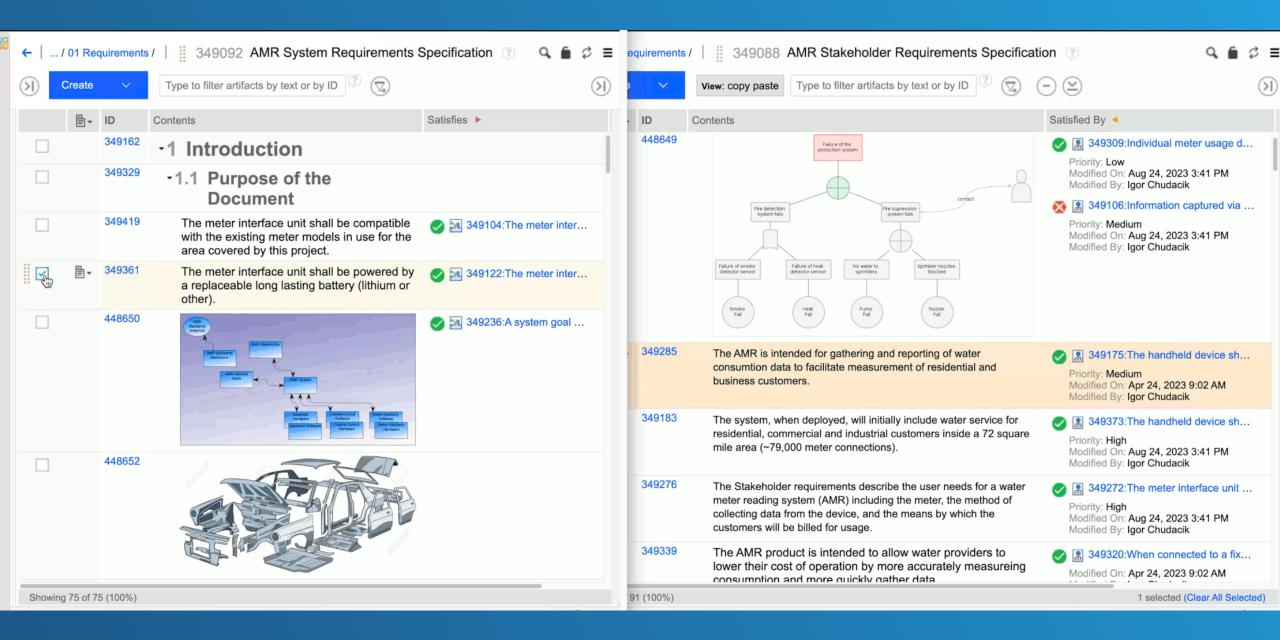
- Duplication of the data
- No reference to the original data

Widely used technique by companies developing big investment project in military area.









### Copy, Paste & Link

Very similar to Copy & Paste. When copying, link of your choice is created between original and copied artifact.

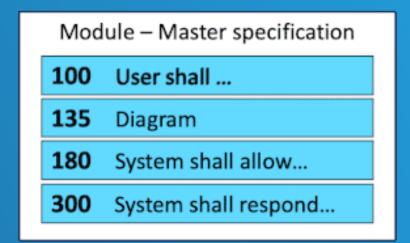
#### **ADVANTAGES**

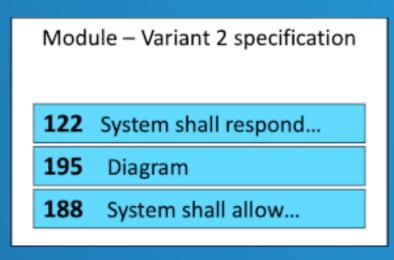
- Live reference to the original artifact
- Link validity can be used to spot the changes between artifacts
- Link + Link Validity = Keeping control of the copy

#### **DISADVANTAGES**

- Duplication of the data
- Some systems do not support copying across different data models

Widely used technique in all industries where you need to track the source requirements during your project.









### Artifact based reuse in specification

The best option for preservation high integrity and single source of truth.

- Artifacts can be reused across several specifications
- Any change to the artifacts in any specification is immediately propagated to all instances in other specifications\*

#### **ADVANTAGES**

Preserving high integrity

#### **DISADVANTAGES**

- Giving up control of the text and attributes = changes are harder to control
- Links, comments and tags may be specification specific, and their changes may not be propagated to other instances of the artifact in other specifications.

Example of simple reuse of hardware requirements specification (often medical and automotive projects)



## **Artifact based Reuse in Specification**

Module – Master specification

100 User shall ...

135 Diagram

**180** System shall allow...

**300** System shall respond...

Module – Variant 2 specification

101 User shall not ...

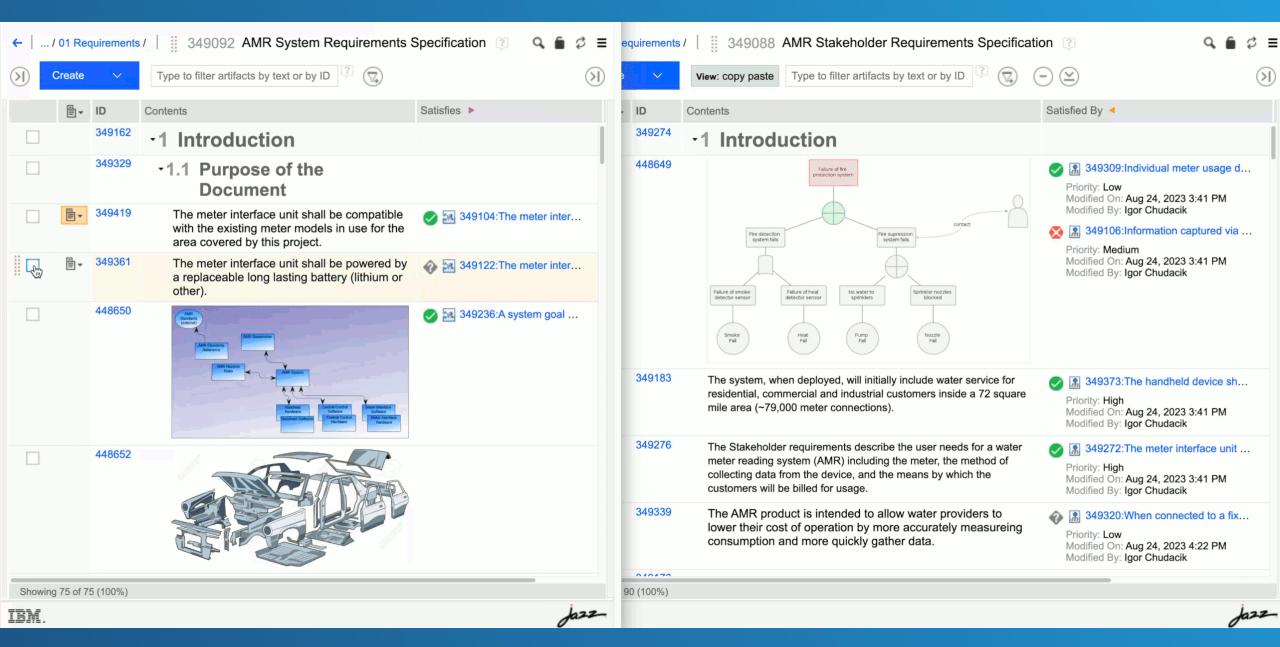
122 System shall respond...

195 Diagram

Module – Variant 4 specification

311 System shall respond...

103 User shall ...

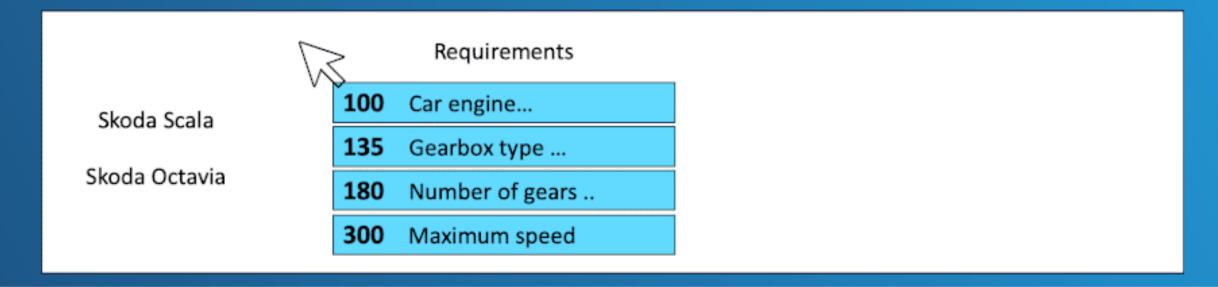


### **Attribute based variants**

Simple method recommended for less complex use-cases. Each variant has associated its specific attributes. The variant is shown based on saved filter.

#### Disadvantage

- Multiplication of attributes
- Traceability relationship links are usually to general requirement not to specific attribute
- Rich text formatting is generally not possible to use for attributes

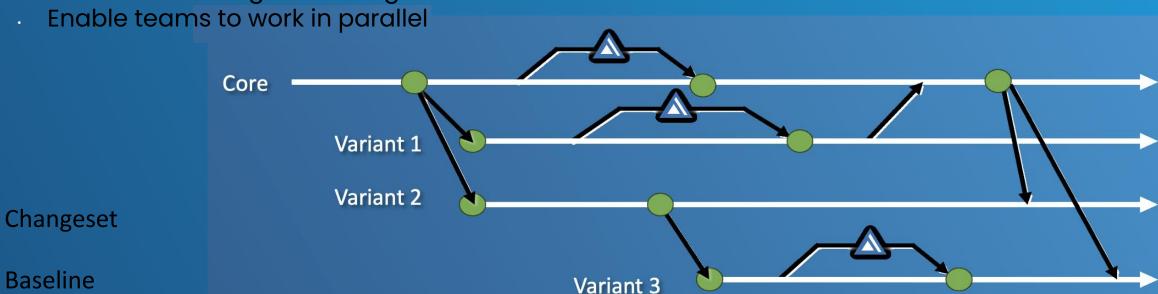


## **Branching & Merging**

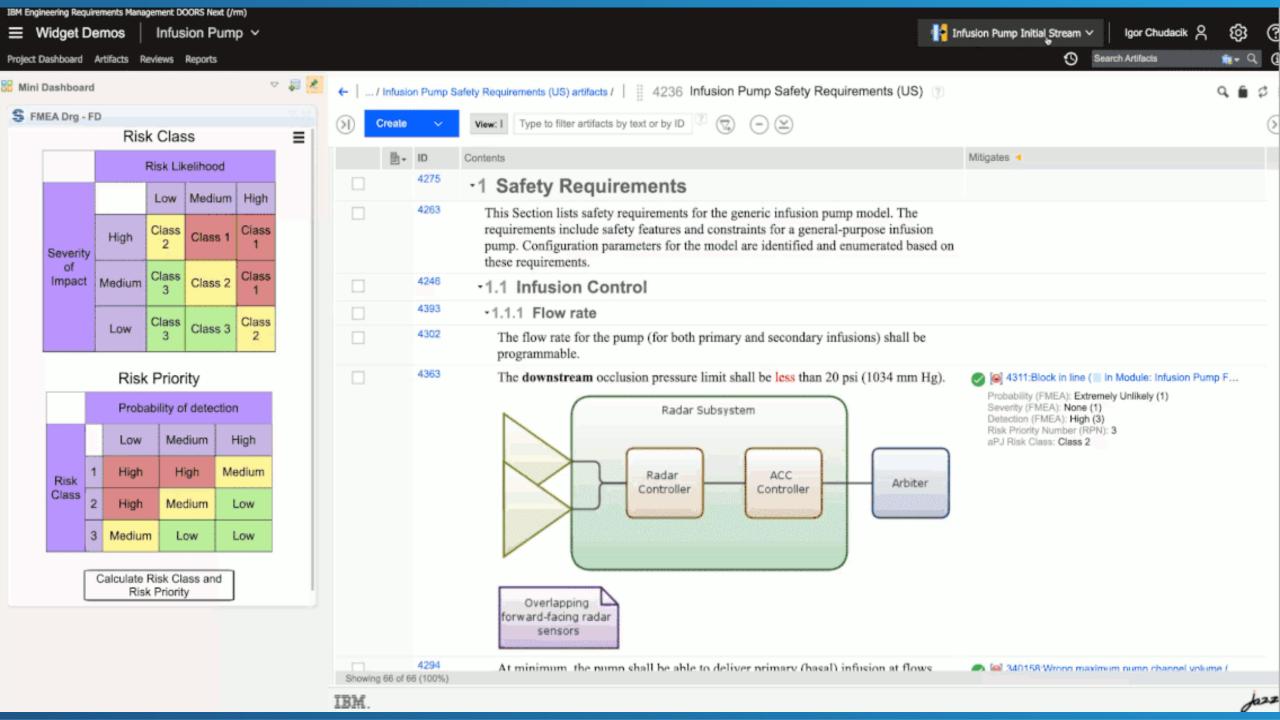
Key concept in systems and software development. It is possible to branch requirements from a component into 2 or more branches or "streams". Streams can be later compared and/or merged.

#### Branch is usually created to:

- Create different variants
- Create work stream for experimental work or new features while having stable release stream
- Control the changes and bug fixes



# **Branching & Merging**



### **Templates**

Project templates - Coping project or component properties

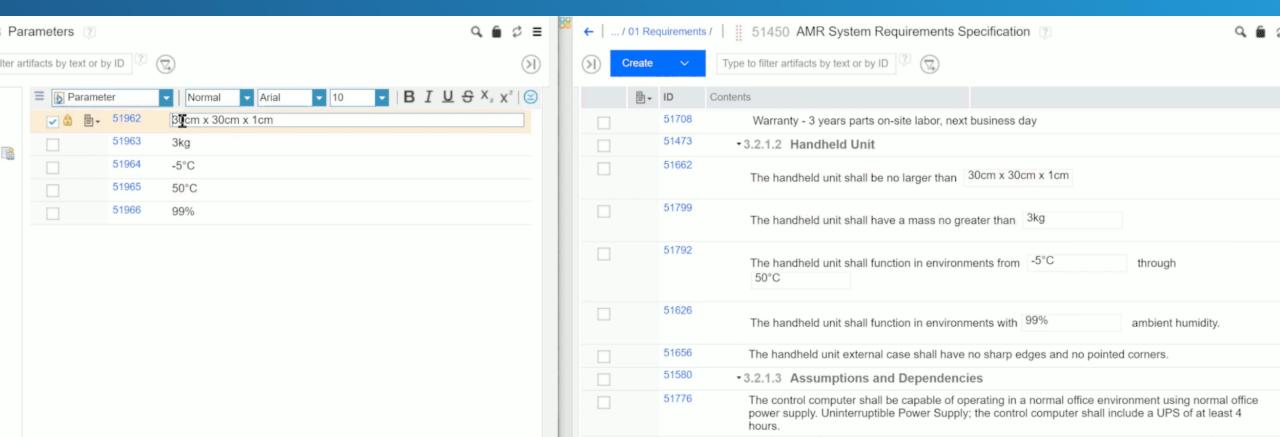
Artifact templates - Reusing only specific artifact

Specification template - Reusing or duplicating all artifact in the specification



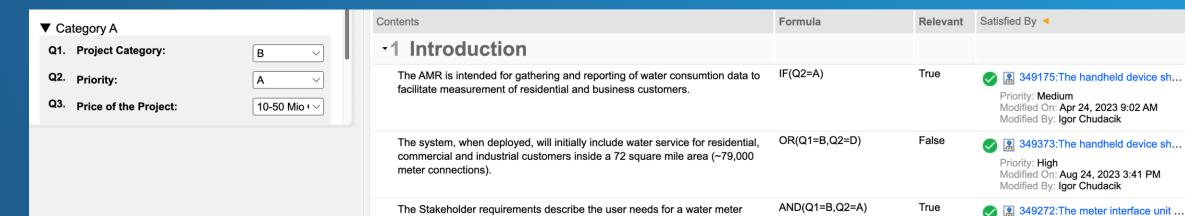
### **Parameters**

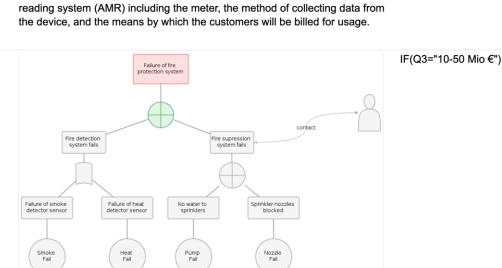
- Parameter values are embedded artifacts
- Parameters are created, changed and stored in separate specification document
- Updating the parameters in specification document will update parameter values in all specifications



### Requirements reuse using logical formulas

- Filter relevant artifact using logical formulas entered by user.
- Filtered artifacts are marked and reused in another specification.
- Every artifact is filter based on its own formula extremely granular.





Priority: High Modified On: Aug 24, 2023 3:41 PM Modified By: Igor Chudacik 349309:Individual meter usage d... Priority: Low Modified On: Aug 24, 2023 3:41 PM Modified By: Igor Chudacik 349106:Information captured via ... Priority: Medium Modified On: Aug 24, 2023 3:41 PM Modified By: Igor Chudacik

Modified On: Apr 24, 2023 9:02 AM Modified By: Igor Chudacik

Modified On: Aug 24, 2023 3:41 PM Modified By: Igor Chudacik

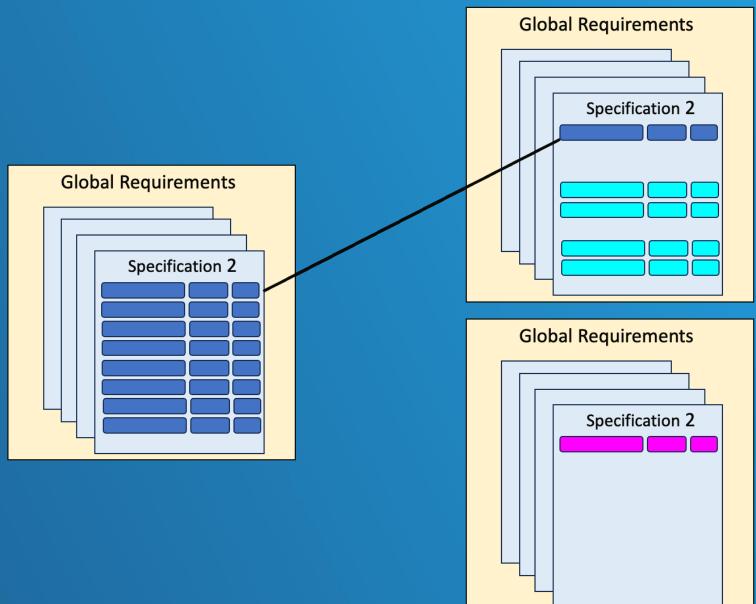
Priority: Medium

Priority: High

True

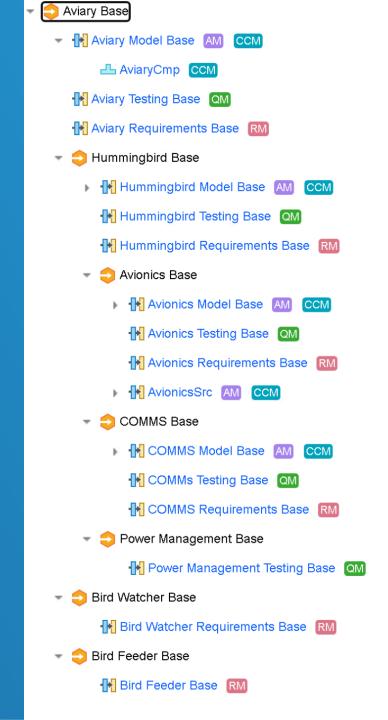
Reuse across components and projects

- Discover, explore and deliver artifact changes across projects, components and streams
- Cherry pick atomic changes all the way to specific attribute



# Reuse of system components

- Most advanced and most powerful form of reuse across entire engineering lifecycle.
- Component represents physical or logical pieces of the system.
  - In requirements management context, component represent set of requirements specifications and artifacts.
  - In a system context, component represents combination of requirements, architecture and tests, organized in hierarchy
- Components make it easy to reuse and combine parts of the system across V model, into different versions, variants or products.
- Disadvantage Increased complexity requires certain level of process maturity in your company, regarding variants management

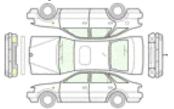


### Reuse of system components

# 000

Standard variant

**Body Standard** 



Rims Standard



**Engine Standard** 



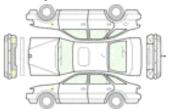


Your product



Variant 1

Body Variant 1



Rims Standard



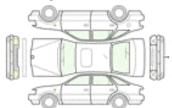
**Engine Standard** 







Body Variant 2



Rims Standard



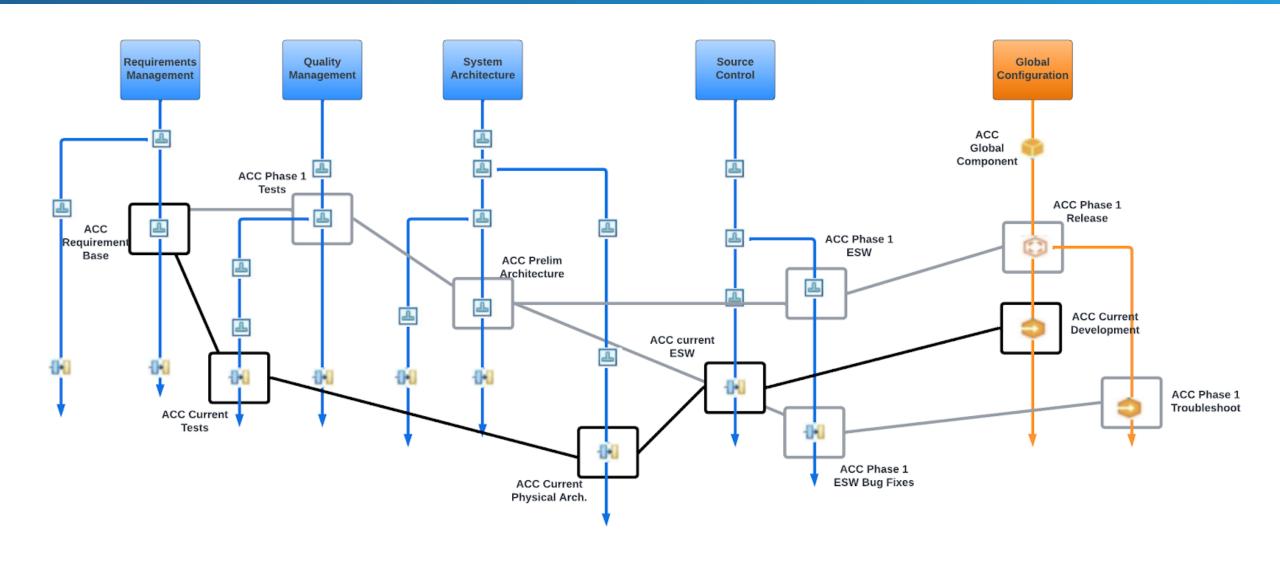
Engine variant 2







## Reuse of system components



### Thank you for your attention

### Get more information at:

- www.softacus.com
- jan.jancar@softacus.com
- andreas.mettauer@softacus.com
- info@softacus.com





# SOFTACUS

**IBM BASED SOLUTIONS** 

