

# Standardisierung mit TIA Portal



# Standardisierung – eine Übersicht

- Standardisierung mit TIA Portal

## Was ist die Anforderung?

- Kürzere Entwicklungszeiten
- Verkürzte Produktlebenszyklen
- Fortschreitende Technikkonvergenz
- Zunehmender globaler Wettbewerb
- Steigende Qualitätsanforderungen

DIN e.V.



## Wie kommt man dahin?

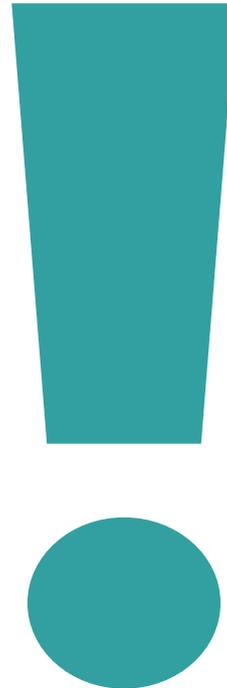
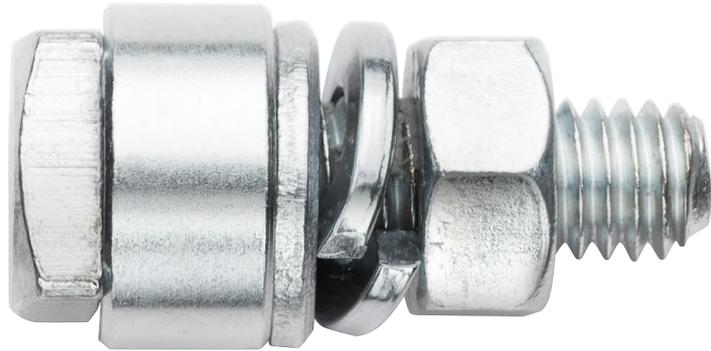
- **Standardisierung** – als Basis der Digitalisierung



# Beispiele für Standardisierung



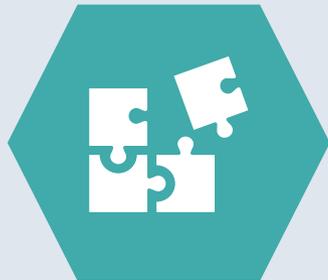
# Beispiele für Standardisierung



**SIEMENS**  
*Ingenuity for life*



Standardisierung  
kann eine Vielzahl  
von Bereichen  
Umfassen



## Geschäftsprozesse

- Bestellung
  - Planung
  - Versand
- 

## Produktionslinien

- Oberflächen
  - Kommunikationskanal
  - Datenstrukturen und Inhalte
  - IT-Integration (Telegramme)
- 

## Maschinen

- Hardware
- Software
- Netzwerk
- Objekte/Module
- Bedienerschnittstelle und Meldungen

# Standardisierung in der TIA Portal Umgebung Engineering

Hardwareplanung	Programmierung	Kommunikation	Safety	Engineering
Diagnose	Bedienen und Beobachten	Technologie	Antriebsintegration	
Verteiltes Arbeiten	Versionierung	Test	Maschinen-simulation	Workflow
Benutzerverwaltung	Lizenzmanagement	Software -managemnet	Sicherheit	Administration

# Agenda

## Zeitplan

09:00	Workshop
10:30	Kaffeepause
10:45	Workshop
12:15	Mittagessen
13:00	Workshop
14:15	Kaffeepause
14:30	Workshop
15:15	Ende

**SIEMENS**  
Ingenuity for Life

**Standardisierung – eine Übersicht**  
- Standardisierung mit TIA Portal

**SIEMENS**  
Ingenuity for Life

**Software Design**  
- Standardisierung mit TIA Portal

**SIEMENS**  
Ingenuity for Life

**Bibliothekskonzept**  
- Standardisierung mit TIA Portal

**SIEMENS**  
Ingenuity for Life

**HMI Design**  
- Standardisierung mit TIA Portal

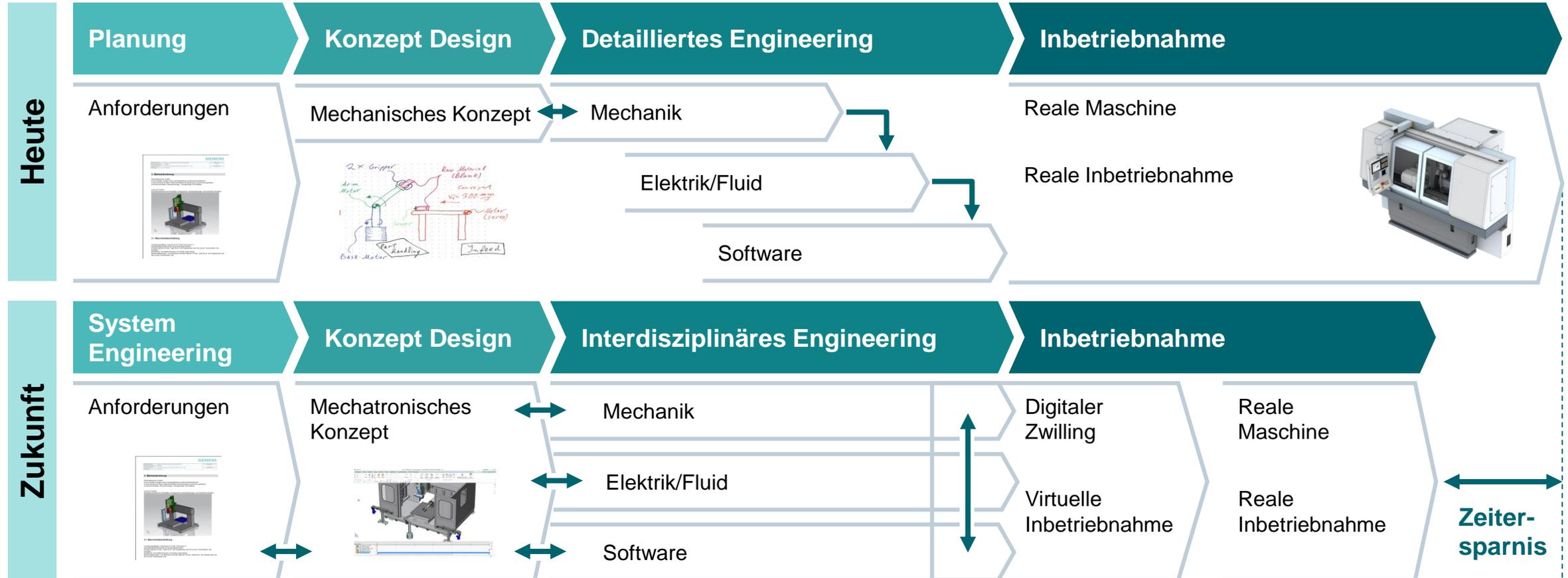
**SIEMENS**  
Ingenuity for Life

**TIA Portal Optionen**  
Arbeiten für und mit Standards

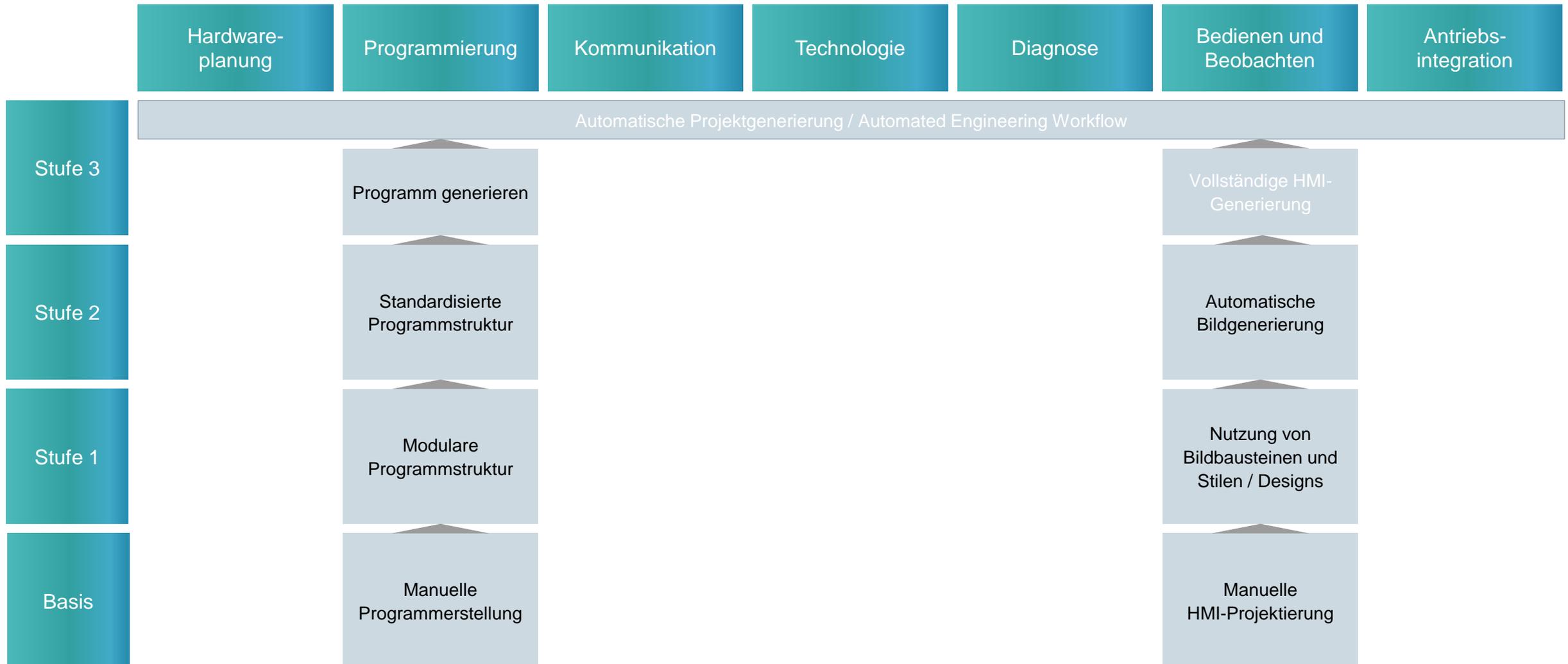
**SIEMENS**  
Ingenuity for Life

**Standardisierungsmatrix**  
Weiteres Interesse an der Standardisierung?

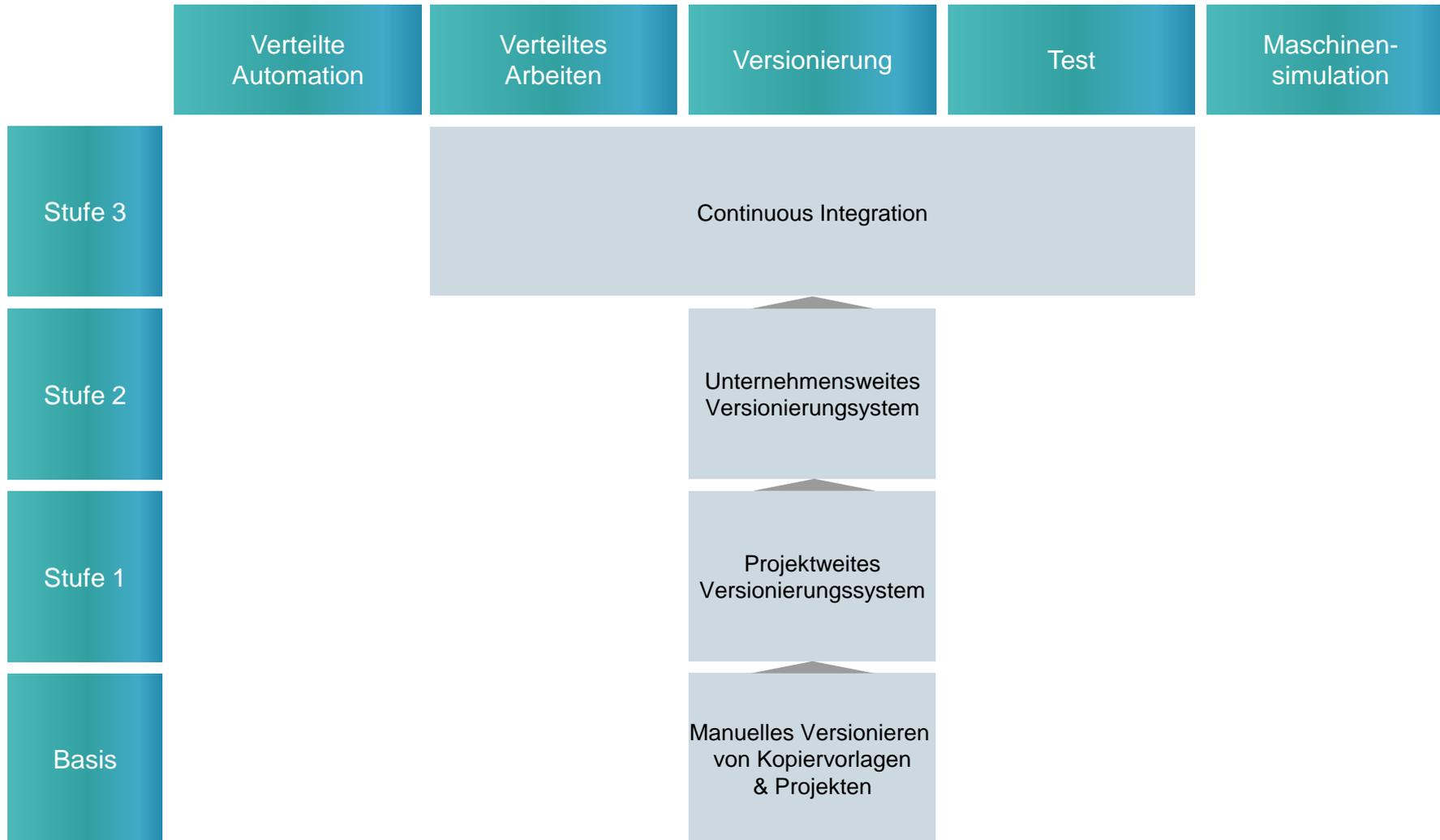
# Standardisierung als Basis der Digitalisierung – Parallelisierung von Arbeitsschritten



# Standardisierungsmatrix in der TIA Portal Umgebung Engineering



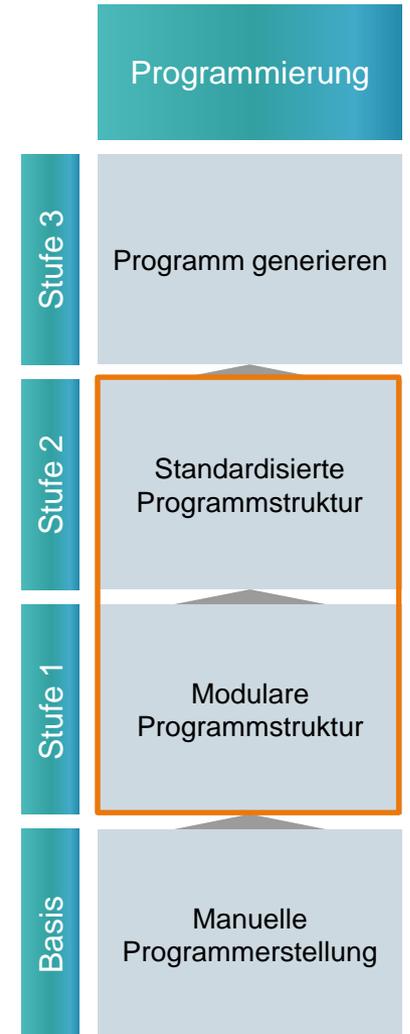
# Standardisierungsmatrix in der TIA Portal Umgebung Engineering Workflow



# Software Design

- Standardisierung mit TIA Portal

**SIEMENS**  
*Ingenuity for life*



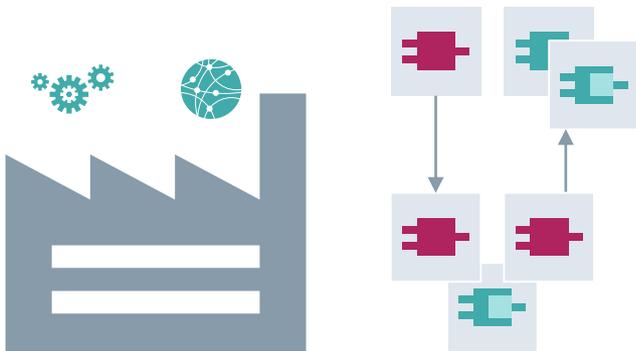
## Standardisierung

Analyse der Maschinenstruktur

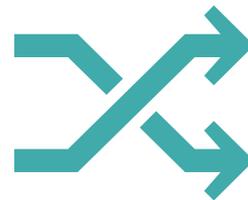
Design der Schnittstellen

Realisierung

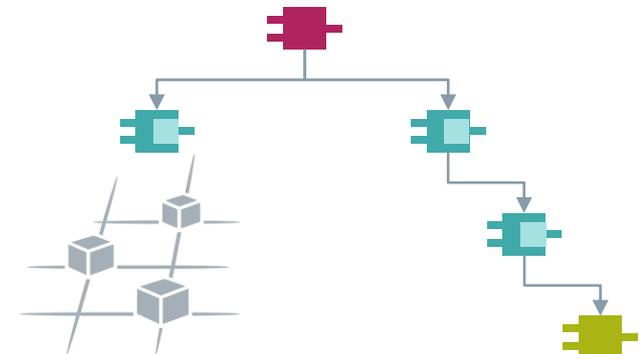
Ausgangssituation



Re-Engineering



Verbesserte und standardisierte Architektur



Standardisierungsleitfaden: <https://support.industry.siemens.com/cs/ww/de/view/109756737>

## Was ist die Herausforderung?

- Sie haben im Jahr **mehrere Projekte/Maschinen mit ähnlichen Funktionen**
- Oftmals wird ein SPS- und HMI-Projekt **manuell** aus alten Projekten **zusammenkopiert**
- Manuelle, kostenintensive, **langwierige Anpassungen** sind jeweils notwendig
- Sie möchten **Kosten** für Anpassungen **reduzieren**



## Wie kommt man dahin?

- **Standardisierung** von Schnittstellen
- **Vereinheitlichung** von Programmierstilen



# Software Design - Werkzeugkasten

# Umsetzung eines Programmierstandards – Erwartungshaltung



# Umsetzung eines Programmierstandards – Programmiersprachen

## Programmiersprachen

- Grafische Sprache
- Textsprache



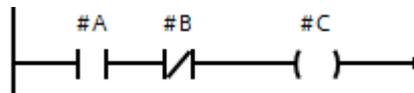
## SPS-Programmiersprachen im TIA-Portal

### Grafische Sprachen

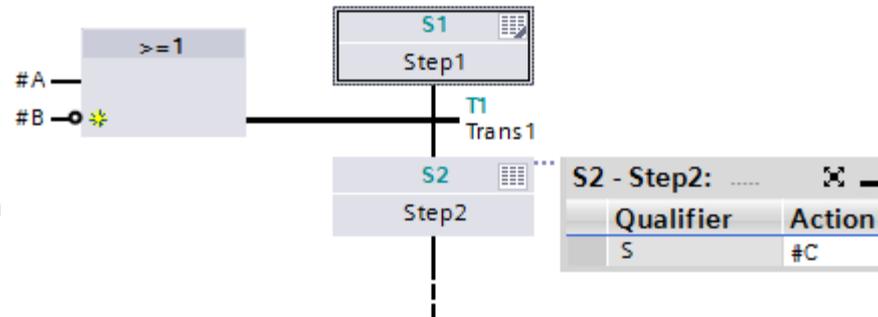
#### FUP



#### KOP



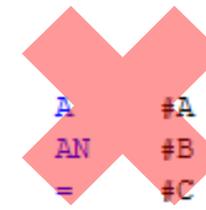
#### GRAPH



- Baustein größtenteils aus Binärverknüpfungen
- Verschaltung / Aufruf von Bausteinen

### Textsprachen

#### AWL



#### SCL

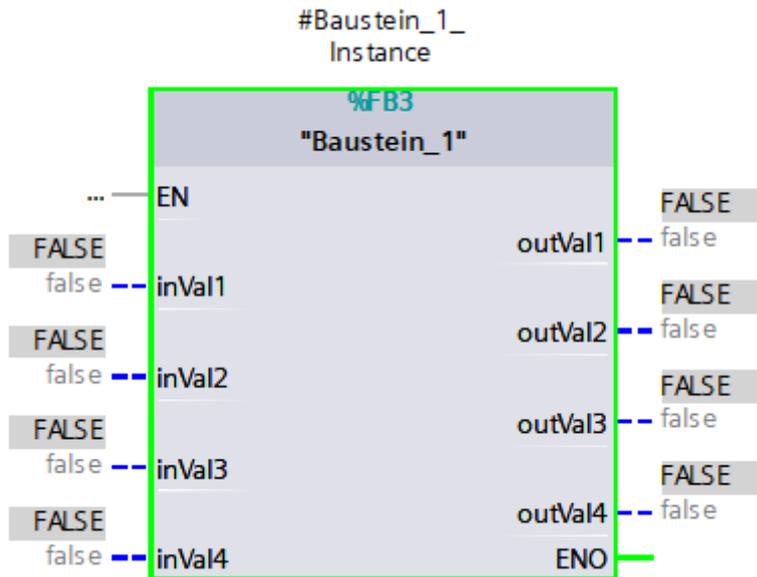
#C := #A AND NOT(#B);

- komplexere Logik
- "Black-Box" für Wartungspersonal

# Programmiersprachen

## Vergleich FUP/SCL – Bausteinaufruf

FUP



SCL

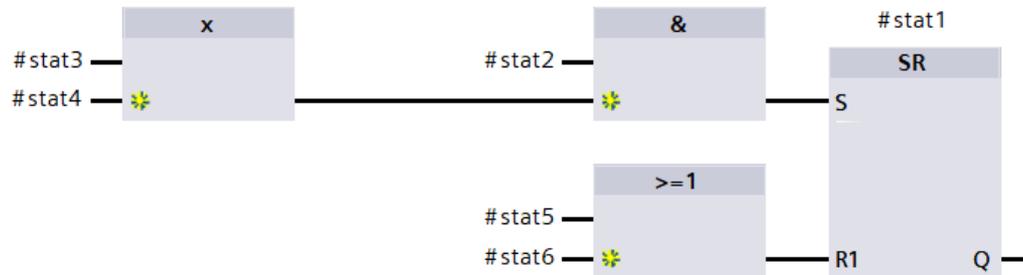
```
#Baustein_1_Instance (inVal1:=#temp1,  
inVal2:=#temp2,  
inVal3:=#temp3,  
inVal4:=#temp4,  
outVal1=>#temp5,  
outVal2=>#temp6,  
outVal3=>#temp7,  
outVal4=>#temp8);
```

#temp1	FALSE
#temp2	FALSE
#temp3	FALSE
#temp4	FALSE
#temp5	FALSE
#temp6	FALSE
#temp7	FALSE
#temp8	FALSE

# Programmiersprachen

## Vergleich FUP/SCL – Binärverknüpfungen

FUP



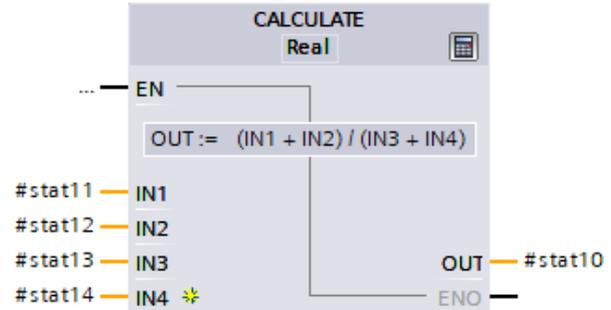
SCL

```
IF (#stat5 OR #stat6) THEN
    #stat1 := FALSE;
ELSIF (#stat2 AND (#stat3 XOR #stat4)) THEN
    #stat1 := TRUE;
END_IF;
```

# Programmiersprachen

## Vergleich FUP/SCL – Mathematische Funktionen

FUP



SCL



```
#stat10 := (#stat11 + #stat12) / (#stat13 + #stat14);
```

# Programmiersprachen

## Vergleich FUP/SCL – Komplexe Implementation

FUP

SCL



```
□CASE #stat10 OF
  1: // Statement section case 1
    ;
  2..4: // Statement section case 2 to 4
    ;
  ELSE // Statement section ELSE
    ;
END_CASE;

□FOR #i := 0 TO #MAX DO
  #avg := #Array[#i] / #MAX;
END_FOR;
```

# Umsetzung eines Programmierstandards – Programmierstyleguide

## Programmierstyleguide

- Variablenbezeichner
- Nomenklatur



# Programmierstyleguide – Bezeichner

- Bezeichner von Variablen beginnen mit **Kleinbuchstaben**
- Kurzer, funktionaler Variablennamen

## Abkürzungen

- Nur eine Abkürzung pro Variablenname

Abk.	Typ
<b>Min</b>	Minimum
<b>Max</b>	Maximum
<b>Act</b>	Actual, Current
<b>Next</b>	Next
<b>Prev</b>	Previous
<b>Avg</b>	Average
<b>Diff</b>	Difference
<b>Pos</b>	Position
<b>Ris</b>	Rising Edge
<b>Fal</b>	Falling Edge
<b>Sim</b>	Simulated
<b>Sum</b>	Sum
<b>Old</b>	Old value (z.B. für Flankenerkennung)

ActVelocityControl			
	Name	Data type	Default value
1	▼ Input		
2	■ startMotor	Bool	false
3	■ setVelocity	Real	10.0
4	■ setAcceleration	Real	0.5
5	▼ Output		
6	■ actualVelocity	Real	0.0
7	■ actualPosition	Real	0.0
8	■ error	Word	16#FFFF
9	▼ InOut		
10	■ ▼ telegram	"typeMotorTelegramm0815"	
11	■ stateWord	Word	
12	■ controlWord	Word	
13	▼ Static		
14	■ statError	Bool	false
15	■ ▶ statVelocityValues	Array[0..#NUMBEROFVALUES] of Real	
16	■ ▶ instCalculatePosition	"CalculatePosition"	
17	▼ Temp		
18	■ tempPosition	Real	
19	▼ Constant		
20	■ NUMBER_OF_VALUES	Int	50
21	■ MIN_VELOCITY	Real	10.0
22	■ MAX_VELOCITY	Real	500.0

# Programmierstyleguide – Strukturen

## Arrays

- Array-Bezeichner ist immer Mehrzahl
- Array-Index beginnt mit 0 und endet mit einer Konstanten

## PLC-Datentypen

- Präfix »type«
- PLC-Datentypen statt Strukturen

## Static- und Temp-Variablen

- Präfix »stat« und »temp«
- Statische Variablen nur lokal aufrufen
- Multiinstanzen mit »inst« kennzeichnen

ActVelocityControl			
	Name	Data type	Default value
1	▼ Input		
2	startMotor	Bool	false
3	setVelocity	Real	10.0
4	setAcceleration	Real	0.5
5	▼ Output		
6	actualVelocity	Real	0.0
7	actualPosition	Real	0.0
8	error	Word	16#FFFF
9	▼ InOut		
10	▼ telegram	"typeMotorTelegramm0815"	
11	stateWord	Word	
12	controlWord	Word	
13	▼ Static		
14	statError	Bool	false
15	▶ statVelocityValues	Array[0..#NUMBEROFVALUES] of Real	
16	▶ instCalculatePosition	"CalculatePosition"	
17	▼ Temp		
18	tempPosition	Real	
19	▼ Constant		
20	NUMBER_OF_VALUES	Int	50
21	MIN_VELOCITY	Real	10.0
22	MAX_VELOCITY	Real	500.0

# Programmierstyleguide – IN/OUT

## Input, Output und InOut

- Keine Präfixe
- Datenaustausch über Schnittstellen
- Viele Variablen als strukturierte Variablen übergeben
- Out-Variablen nur 1-mal pro Zyklus schreiben

ActVelocityControl			
	Name	Data type	Default value
1	▼ Input		
2	startMotor	Bool	false
3	setVelocity	Real	10.0
4	setAcceleration	Real	0.5
5	▼ Output		
6	actualVelocity	Real	0.0
7	actualPosition	Real	0.0
8	error	Word	16#FFFF
9	▼ InOut		
10	▼ telegram	"typeMotorTelegramm0815"	
11	stateWord	Word	
12	controlWord	Word	
13	▼ Static		
14	statError	Bool	false
15	▶ statVelocityValues	Array[0..#NUMBEROFVALUES] of Real	
16	▶ instCalculatePosition	"CalculatePosition"	
17	▼ Temp		
18	tempPosition	Real	
19	▼ Constant		
20	NUMBER_OF_VALUES	Int	50
21	MIN_VELOCITY	Real	10.0
22	MAX_VELOCITY	Real	500.0

# Programmierstyleguide – Konstanten

## Konstanten

- Namen von Konstanten immer in GROSSSCHRIFT
- Nur lokale Konstanten verwenden
- Konstanten bei Wertabfragen ungleich 0 verwenden: Fehlercodes, ...

ActVelocityControl			
	Name	Data type	Default value
1	▼ Input		
2	startMotor	Bool	false
3	setVelocity	Real	10.0
4	setAcceleration	Real	0.5
5	▼ Output		
6	actualVelocity	Real	0.0
7	actualPosition	Real	0.0
8	error	Word	16#FFFF
9	▼ InOut		
10	▼ telegram	"typeMotorTelegramm0815"	
11	stateWord	Word	
12	controlWord	Word	
13	▼ Static		
14	statError	Bool	false
15	statVelocityValues	Array[0..#NUMBEROFVALUES] of Real	
16	instCalculatePosition	"CalculatePosition"	
17	▼ Temp		
18	tempPosition	Real	
19	▼ Constant		
20	NUMBER_OF_VALUES	Int	50
21	MIN_VELOCITY	Real	10.0
22	MAX_VELOCITY	Real	500.0

# Programmierstyleguide – Spickzettel

Bezeichner	Global – start	Konstanten – MAX_SPEED
Präfix	–	–
Anfangsbuchstabe	klein	–
Schreibweise	camelCasing	GROSSBUCHSTABEN

Formalparameter	In – enable	Out – done	InOut – motorDATA	Stat – statVelocity	Temp – tempVelocity	Constants – MAX_VELOCITY
Präfix	–	–	–	stat'	temp'	–
Anfangsbuchstabe	klein	klein	klein	klein	klein	–
Schreibweise	camelCasing	camelCasing	camelCasing	camelCasing	camelCasing	GROSSBUCHSTABEN

Datenbausteine	Global – MotorData	Einzelinstanz – InstHeater	Multiinstanz – instMotor
Präfix	–	'Inst'	'inst'
Anfangsbuchstabe	Groß	Groß	klein
Schreibweise	PascalCasing	PascalCasing	camelCasing

	Präfix	Anfangsbuchstabe	Schreibweise
<b>OB Organisationsbaustein</b>	StationMain	–	Groß PascalCasing
<b>FB Funktionsbaustein</b>	ConveyorControl	–	Groß PascalCasing
<b>FC Funktion</b>	Filling	–	Groß PascalCasing
<b>TO Technologieobjekt</b>	PositioningAxis	–	Groß PascalCasing
<b>PLC Datentypen</b>	typeMotor	–	klein camelCasing
<b>Beobachtungstabelle</b>	MotorTags	–	Groß PascalCasing
<b>PLC Variablen-tabelle</b>	InputTags	–	Groß PascalCasing
<b>Traces</b>	SpeedAxis	–	Groß PascalCasing
<b>Messungen</b>	HeaterControl	–	Groß PascalCasing

## Basis

- Einzigartige, verständliche Kennungen in Englisch
- Keine Sonderzeichen (\$%& ...)
- Maximal 24 Zeichen
- Arrays: axesData (0 ... MAX) of type

## Abkürzungen<sup>1</sup> (nur eine pro Kennung)

<b>Min/Max</b>	Minimum/Maximum
<b>Act</b>	Aktuell
<b>Next/Prev</b>	Nächster/Vorheriger
<b>Avg</b>	Durchschnitt
<b>Diff/Sum</b>	Differenz/Summe
<b>Pos</b>	Position
<b>Ris/Fal</b>	Steigende/Fallende Signalfanke
<b>Sim</b>	Simuliert
<b>Old</b>	Alter Wert
<b>Dir</b>	Richtung
<b>Err</b>	Fehler
<b>Warn</b>	Warnung
<b>Cmd</b>	Befehl

1 Empfehlungen

# Umsetzung eines Programmierstandards – Schnittstellendefinition



## Schnittstellendefinition

- Modularisierung
- Informationsaustausch

# Schnittstellendefinition – Modularisierung nach DIN EN 61512-1

## Unit (de: **Teilanlage**)

Die oberste Ebene der Gruppierung. Sie umfasst zusammengehörende Einzelsteuereinheiten und technische Einrichtungen, in denen größere Verarbeitungsaktivitäten ausgeführt werden

---

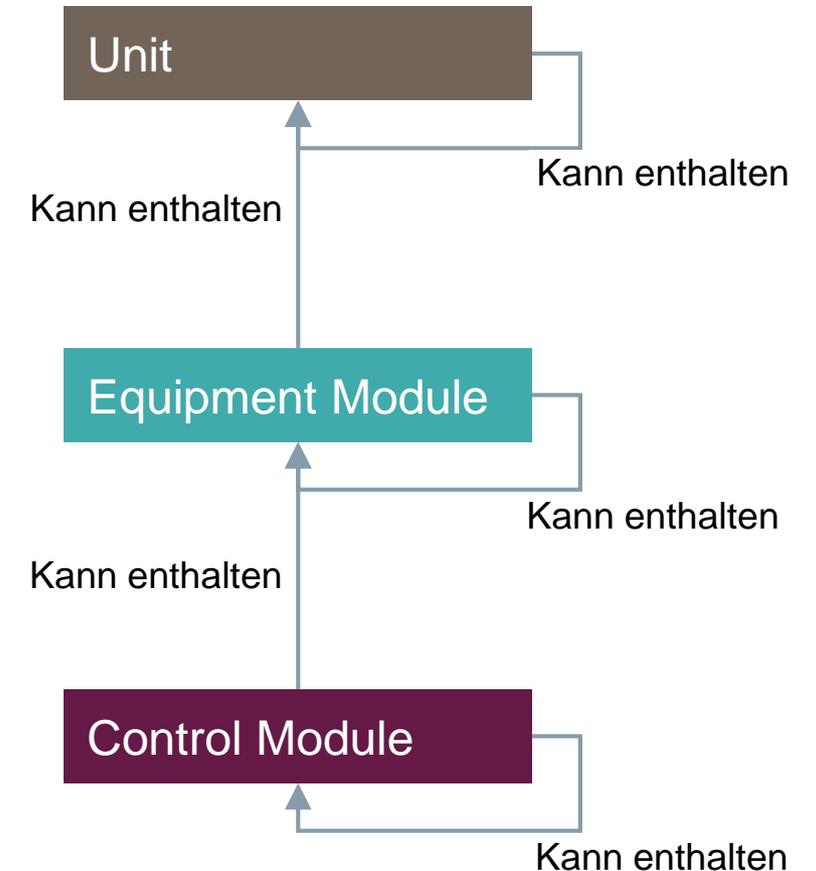
## Equipment Module (de: **Technische Einrichtung**)

Eine funktionale Zusammenfassung von Einzelsteuereinheiten, die eine endliche Anzahl von bestimmten kleineren Verarbeitungsaktivitäten ausführen kann

---

## Control Module (de: **Einzelsteuereinheit**)

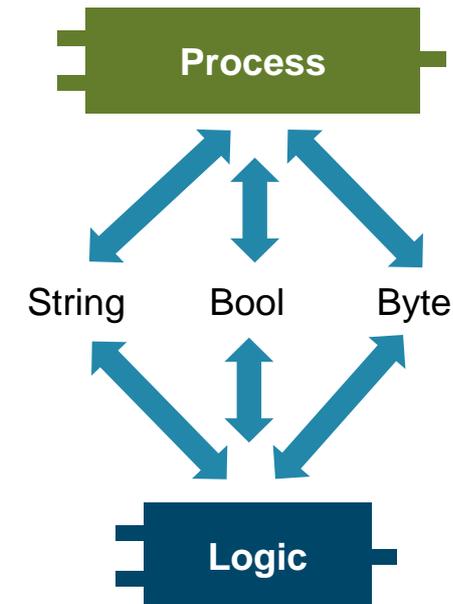
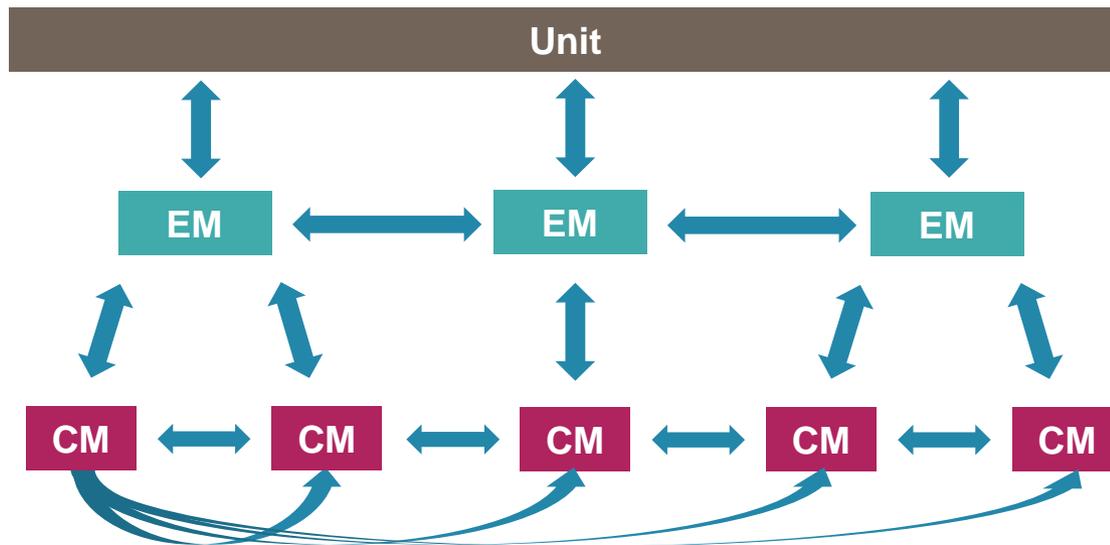
Die unterste Ebene der Gruppierung von Einrichtungen  
→ **Die Basisautomatisierung**



# Schnittstellendefinition – Modularisierung nach DIN EN 61512-1

1. Wer tauscht mit wem Informationen aus?

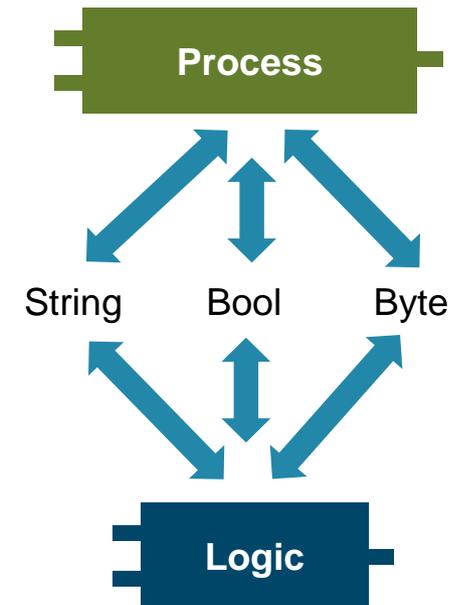
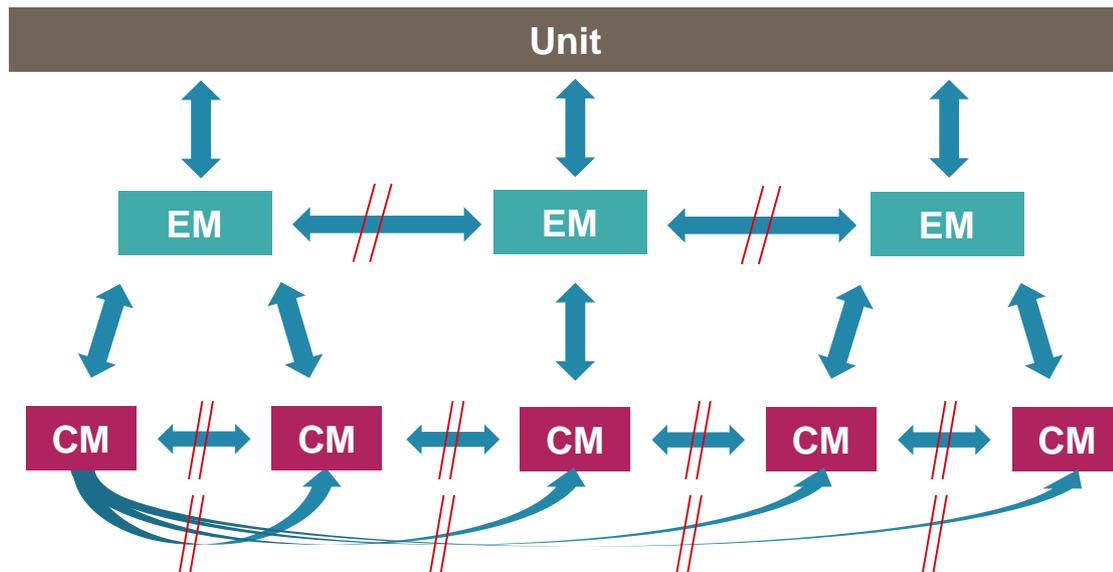
2. Welche Informationen sollen ausgetauscht werden?



# Schnittstellendefinition – Modularisierung nach DIN EN 61512-1

1. Wer tauscht mit wem Informationen aus?

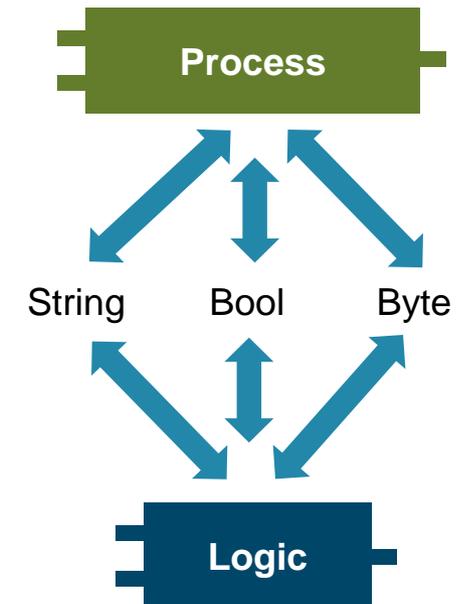
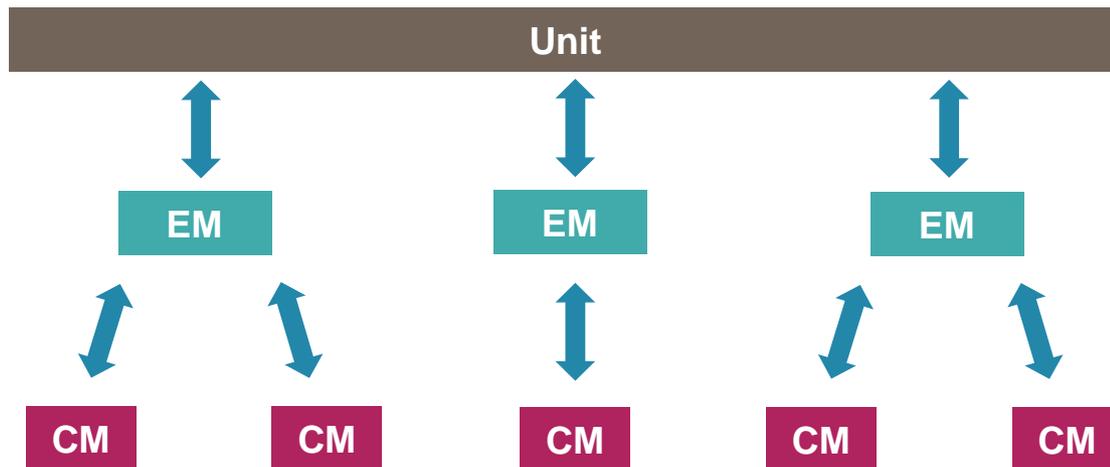
2. Welche Informationen sollen ausgetauscht werden?



# Schnittstellendefinition – Modularisierung nach DIN EN 61512-1

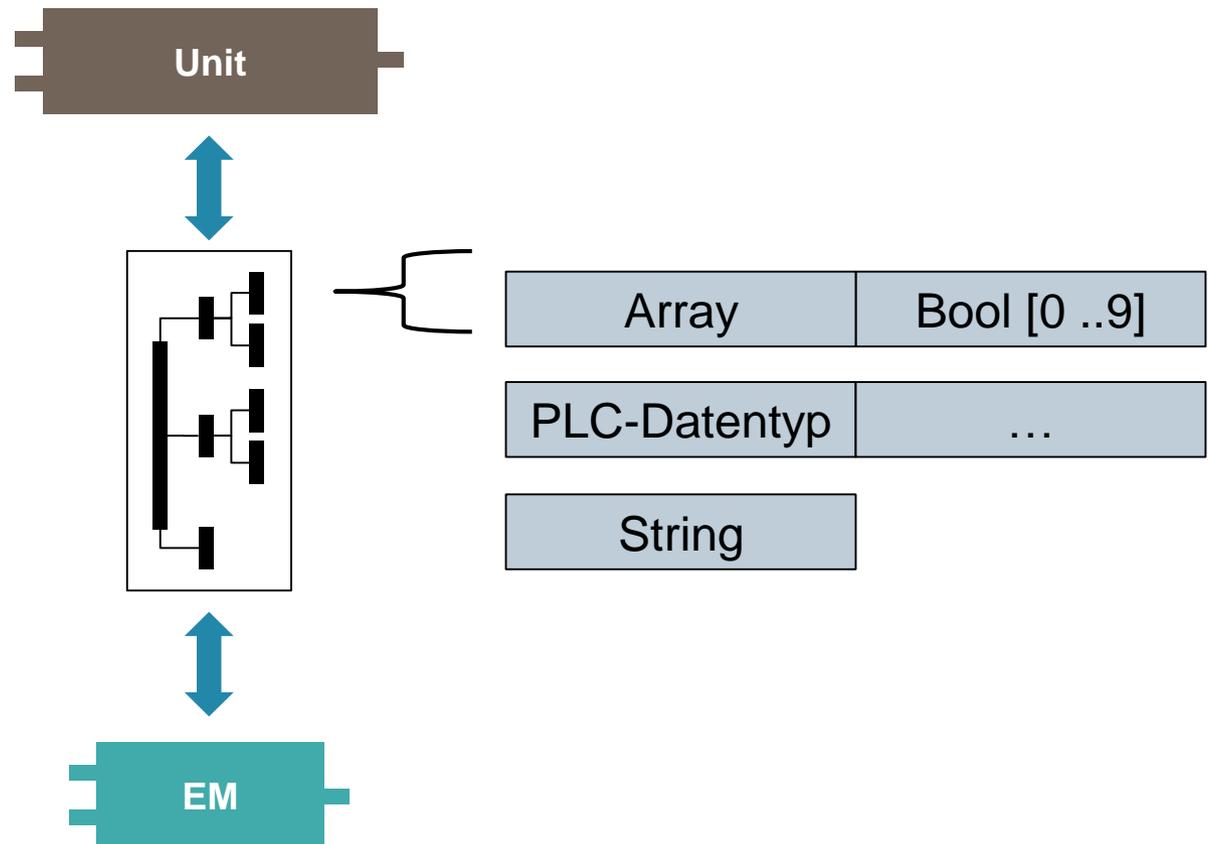
## 1. Wer tauscht mit wem Informationen aus?

## 2. Welche Informationen sollen ausgetauscht werden?



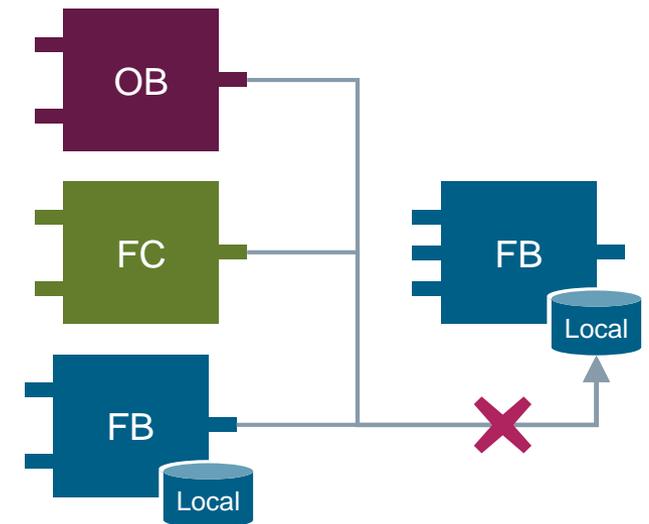
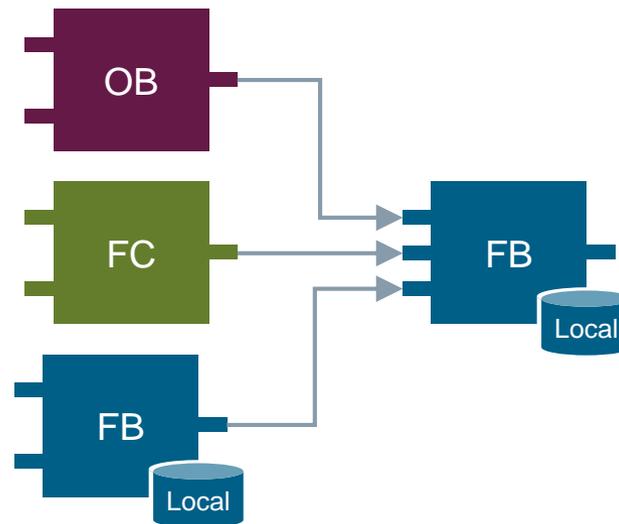
# Schnittstellendefinition – PLC-Datentyp

- Änderungen am Datentyp werden an allen Verwendungsstellen aktualisiert
- Vermeidung von unbeabsichtigt falscher Verschaltung von Schnittstellen
- Vereinfachter Datenaustausch über Bausteinschnittstellen



# Schnittstellendefinition – Parameterübergabe

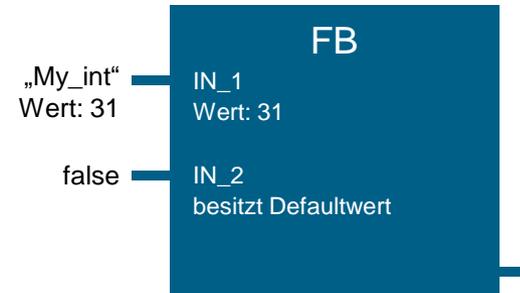
- Programm kann modular, flexibel aus fertigen Bausteinen zusammengesetzt werden
- Programm ist einfach erweiterbar und wartbar
- Programmcode ist leichter lesbar und testbar, da es keine verdeckten Querzugriffe gibt



# Schnittstellendefinition – Parameterübergabe

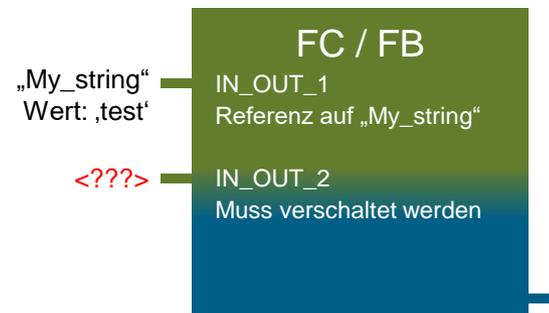
## Call-by-value

- Erzeugt eine Kopie des Wertes
- Benötigt zusätzlichen Arbeitsspeicher



## Call-by-reference

- Erzeugt direkten Verweis auf Speicherplatz des Werts
- Benötigt keinen zusätzlichen Arbeitsspeicher
- Manipulation durch höhere OB-Priorität möglich



Baustein/ Schnittstelle	Elementarer Datentyp	Strukturierter Datentyp
FC	Input	Kopie
	Output	Kopie
	Input/Output	Kopie
FB	Input	Kopie
	Output	Kopie
	Input/Output	Kopie
		<b>Referenz</b>

# Umsetzung eines Programmierstandards – Modularität

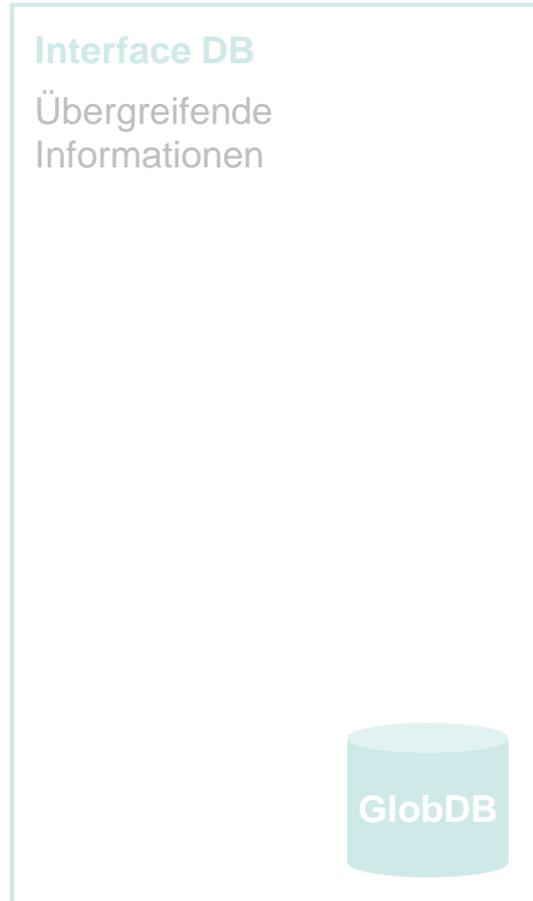


## Modularität

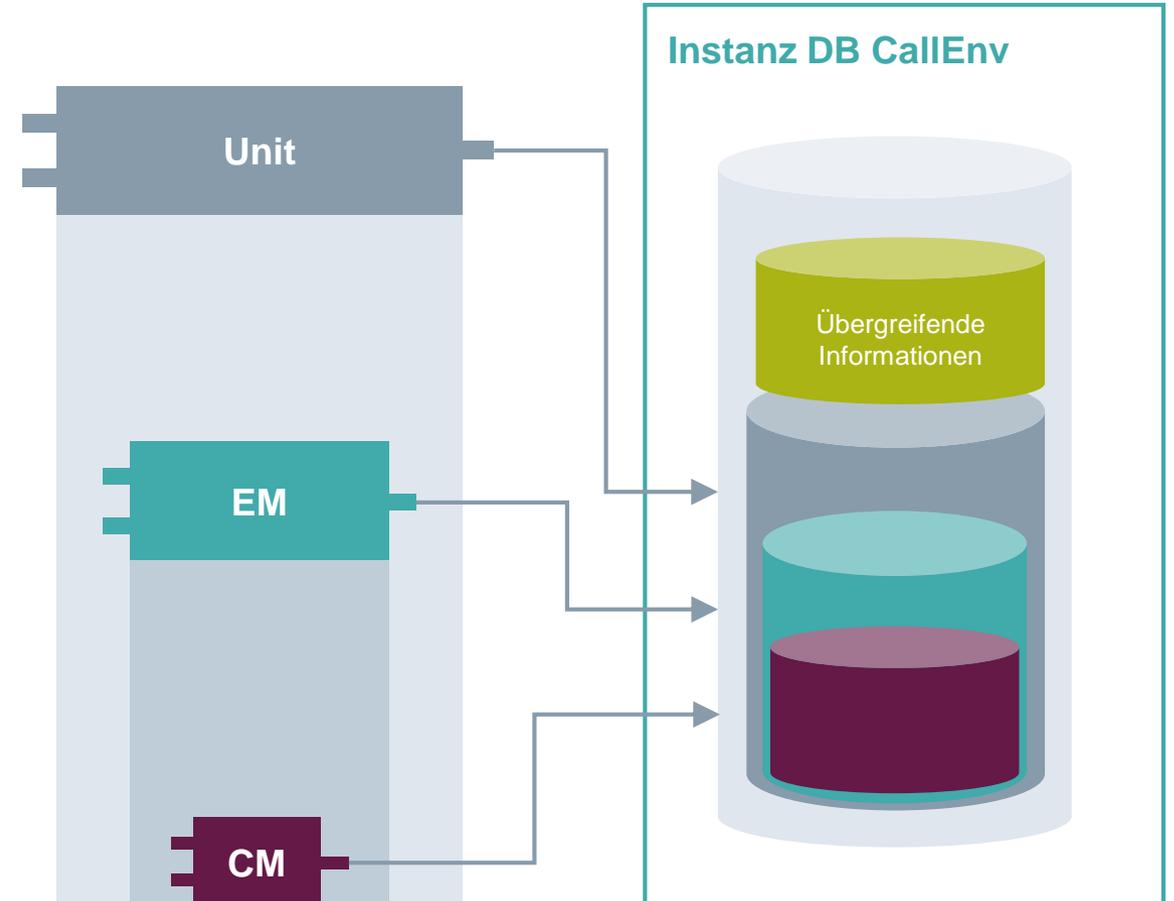
- Speicherablagekonzept

# Modularität – Speicherablage in Global- und Multiinstanz-Datenbausteinen

## Einzelinstanz-Datenbausteine

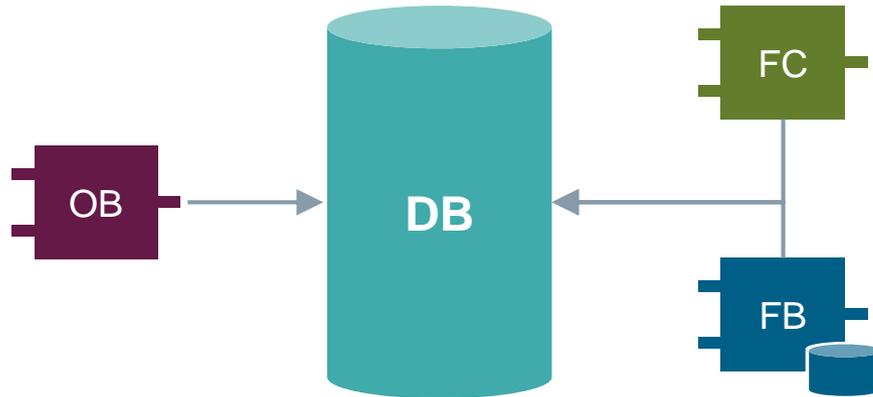


## Multiinstanz-Datenbausteine



# Modularität – Speicherablage in Global- und Multiinstanz-Datenbausteinen

## Global-Datenbausteine



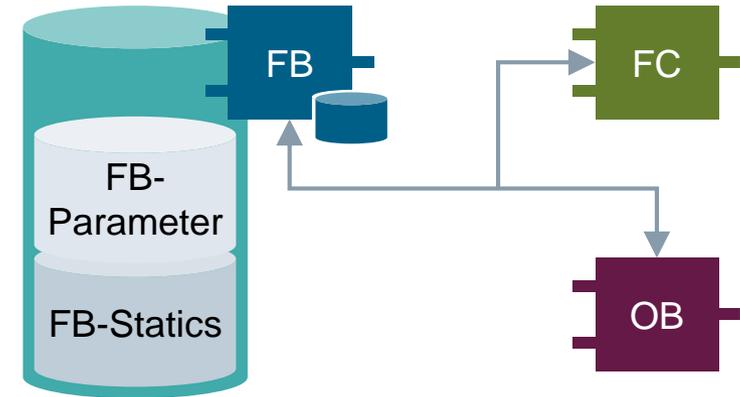
### ✓ Vorteile

Zentrale Datenablage

### ✗ Nachteile

- Unüberschaubare Querzugriffe können auftreten
- Dateninkonsistenzen können auftreten

## Multiinstanz-Datenbausteine



### ✓ Vorteile

Kopieren von Programmen vereinfacht (keine Querzugriffe)

### ✗ Nachteile

Schnittstelle des FBs wird komplexer (Informationsweitergabe)



# Software Design - Beispiele

# Standardisierung an Hand eines vereinfachten Beispiels – (1) Analyse / Funktionaler Schnitt

## Maschine – Säge

Säge



## Maschine – Fräse

Fräse



## Maschine – Bohrer

Bohrer

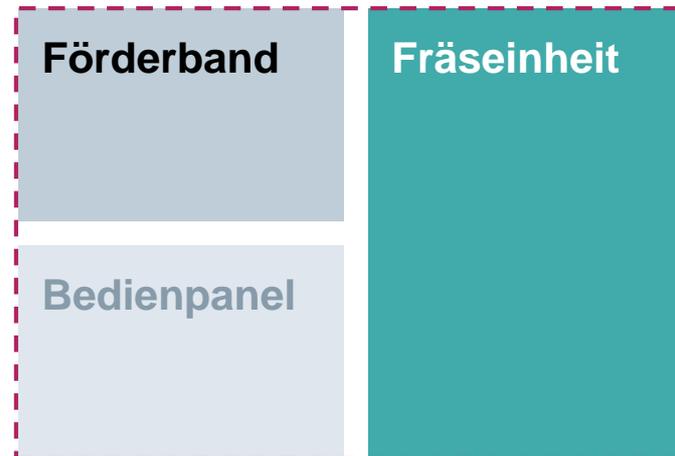


# Standardisierung an Hand eines vereinfachten Beispiels – (1) Analyse / Funktionaler Schnitt

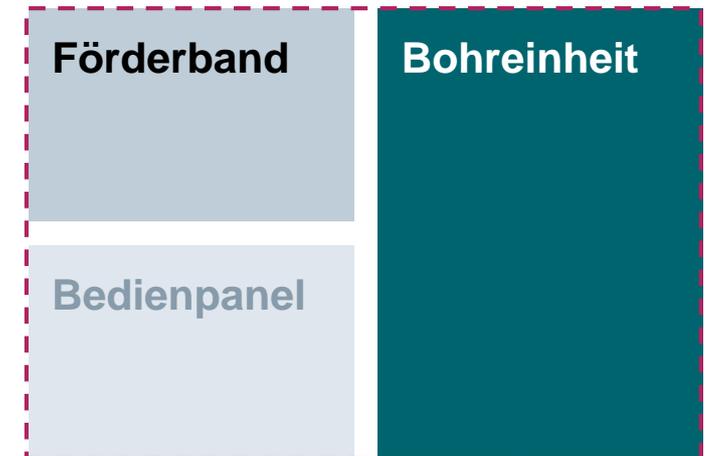
## Maschine – Säge



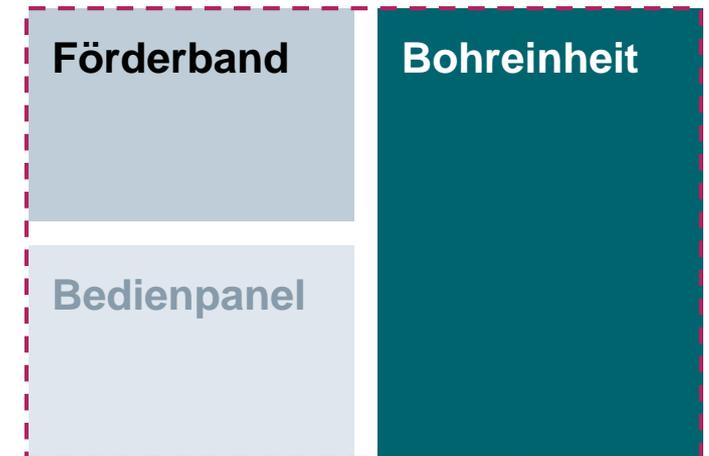
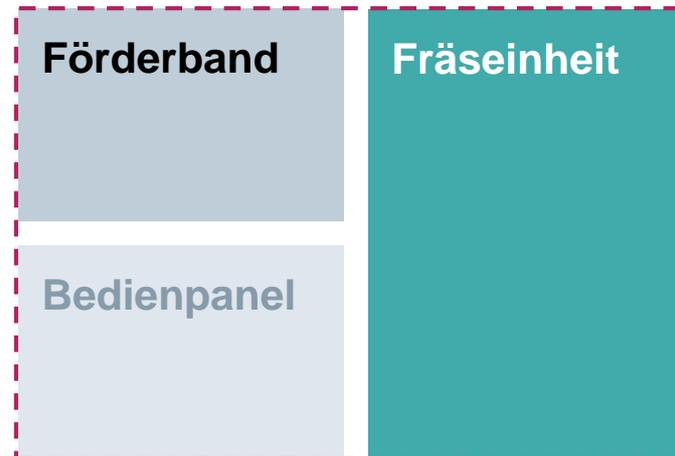
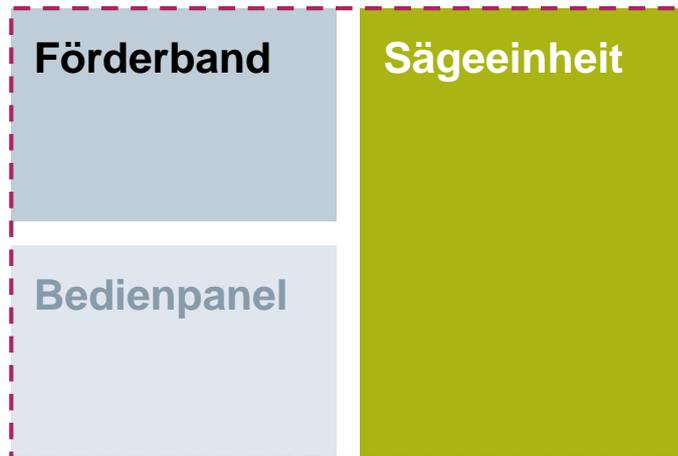
## Maschine – Fräse



## Maschine – Bohrer



# Standardisierung an Hand eines vereinfachten Beispiels – (2) Modularisierung



# Standardisierung an Hand eines vereinfachten Beispiels – (2) Modularisierung

## Standardmodule

**Förderband**

**Bedienpanel**

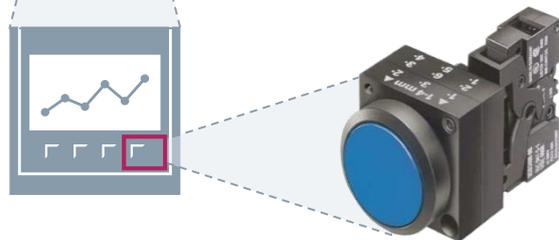
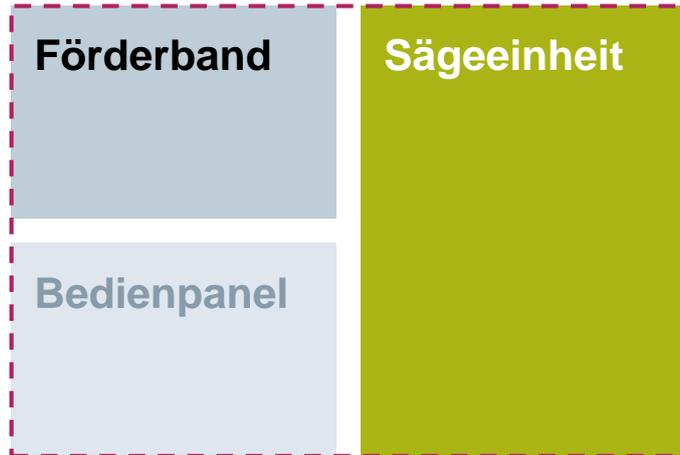
**Sägeinheit**

**Fräseinheit**

**Bohreinheit**

Entwicklung: Einmal  
Instanziierung: Je 3 mal

# Standardisierung an Hand eines vereinfachten Beispiels – (3) Schnittstellendefinition



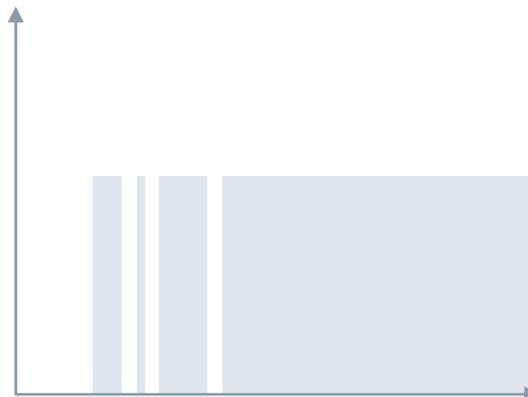
Die Hauptarbeit bei der Entwicklung und Konstruktion eines Standards besteht in der Schnittstellendefinition



Schnittstellen sollten möglichst unabhängig von der **Kundenanforderung** und dem jeweiligen **Modul** sein



Sind die Schnittstellen definiert können die Module parallel entwickelt und getestet werden



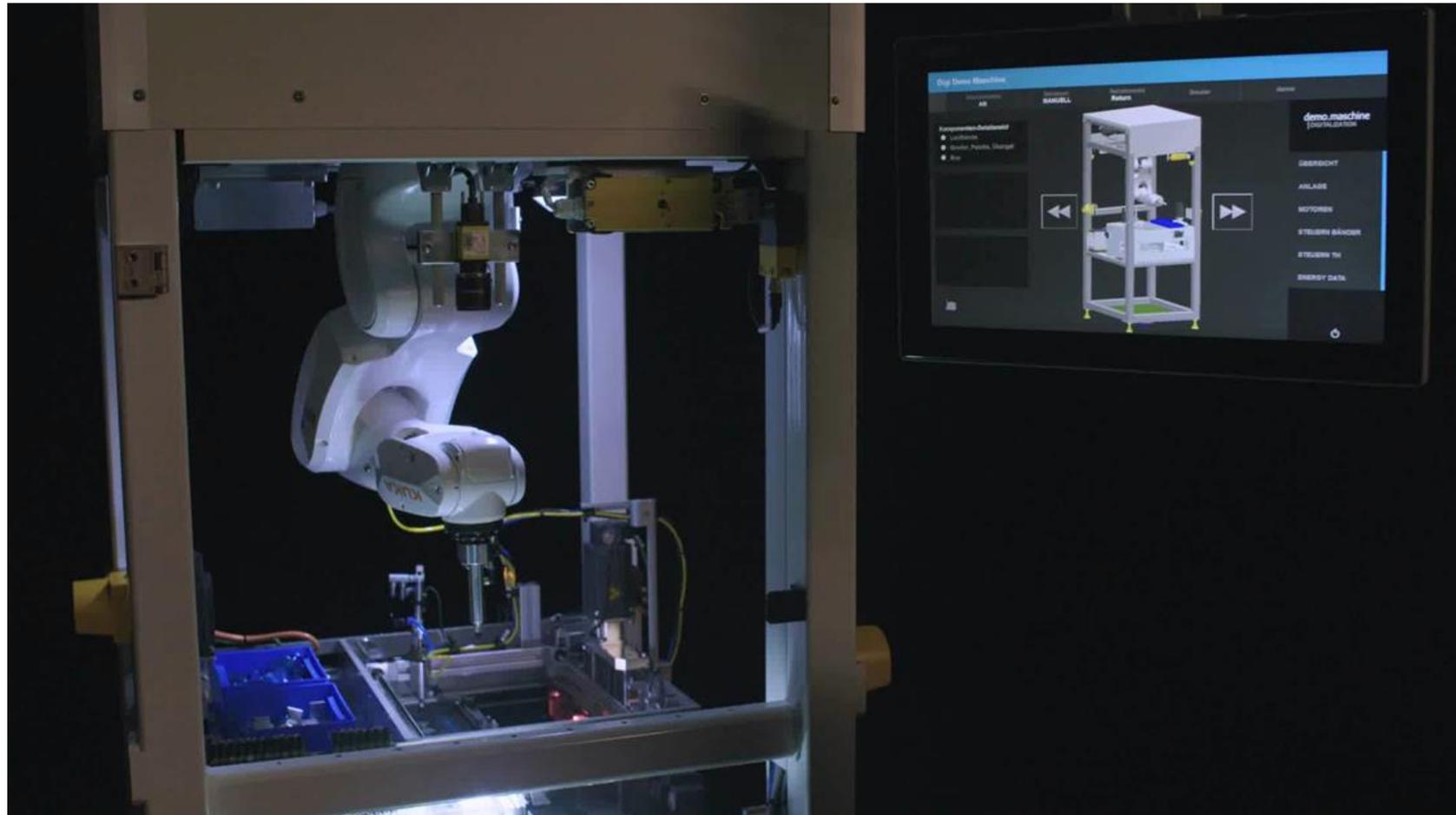
## Funktionen

- Entprellen
- Flanken
- Signal
- Invertiertes Signal

■ ▶ TasterEin	"CBas_typeCmBinSignalStatus"
■ ▼ TasterAus	"CBas_typeCmBinSignalStatus"
■ Q	Bool
■ QBad	Bool
■ Ris	Bool
■ Fal	Bool

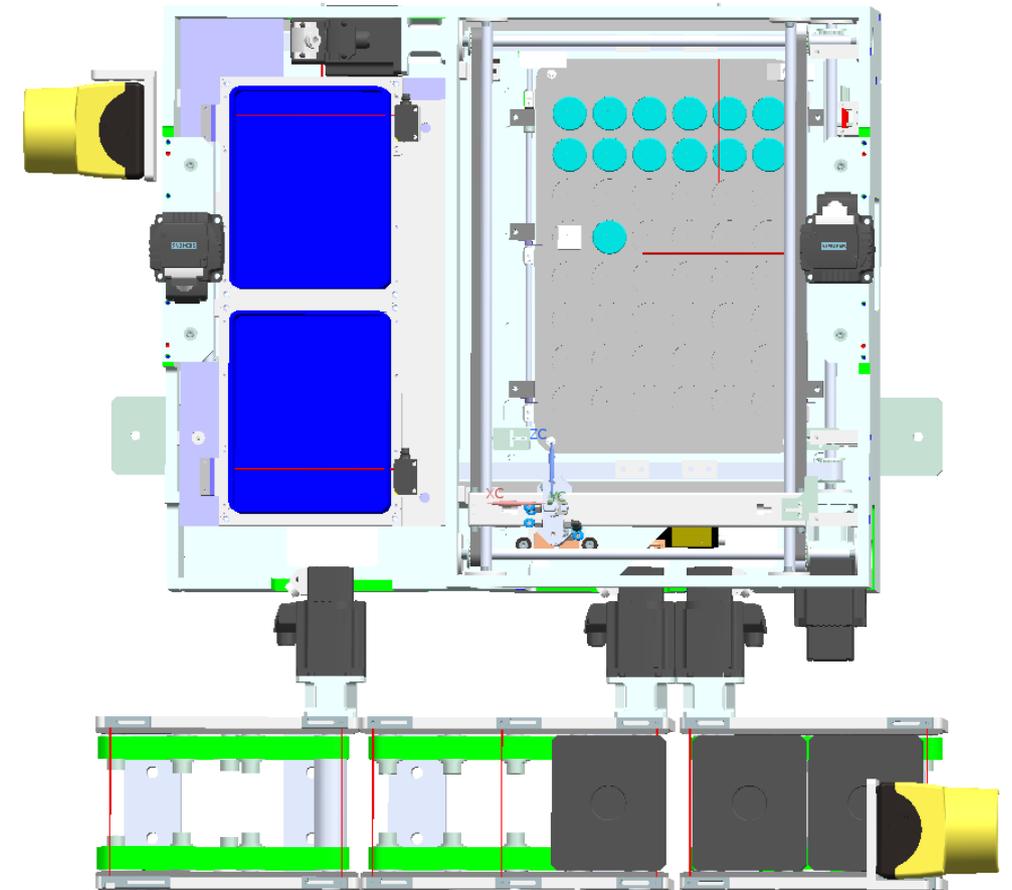
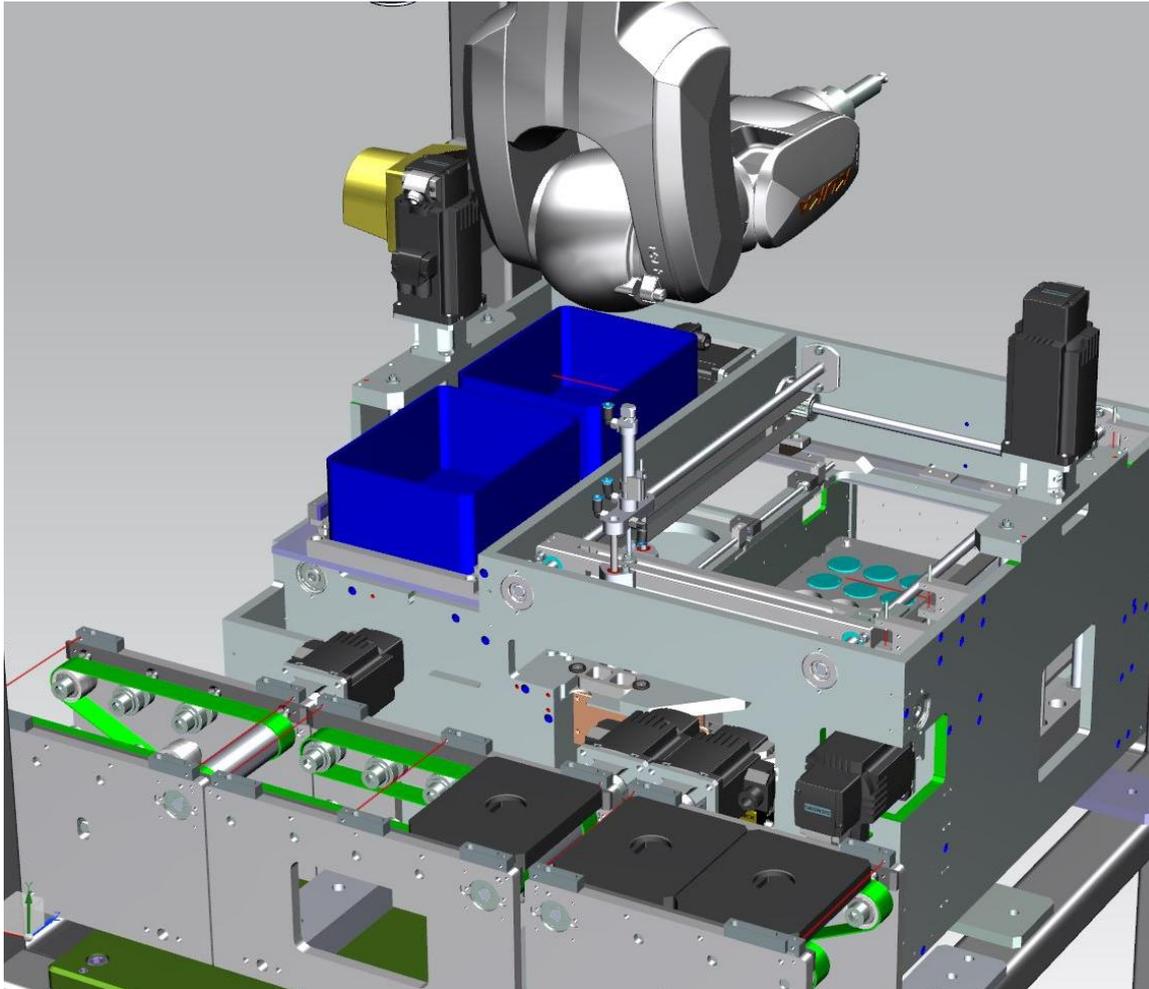
# Standardisierung an Hand eines Beispiels – TIA Digitalization Demonstrator

**SIEMENS**  
*Ingenuity for life*

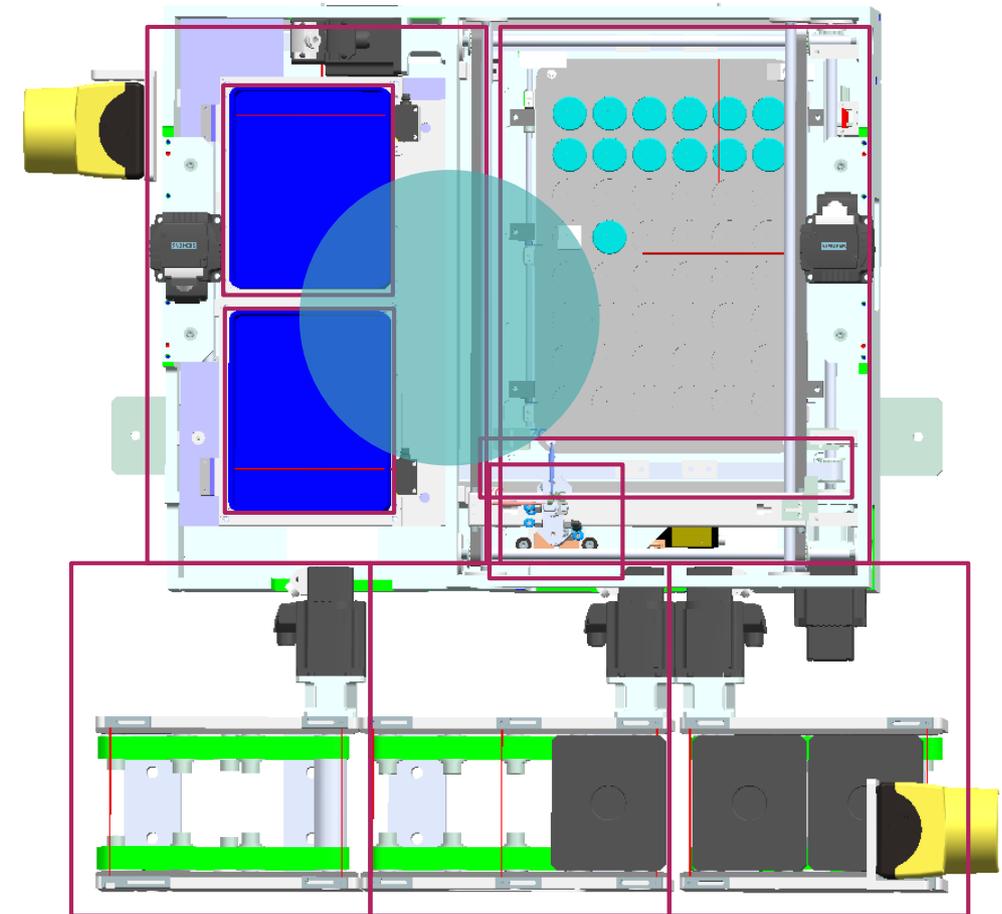
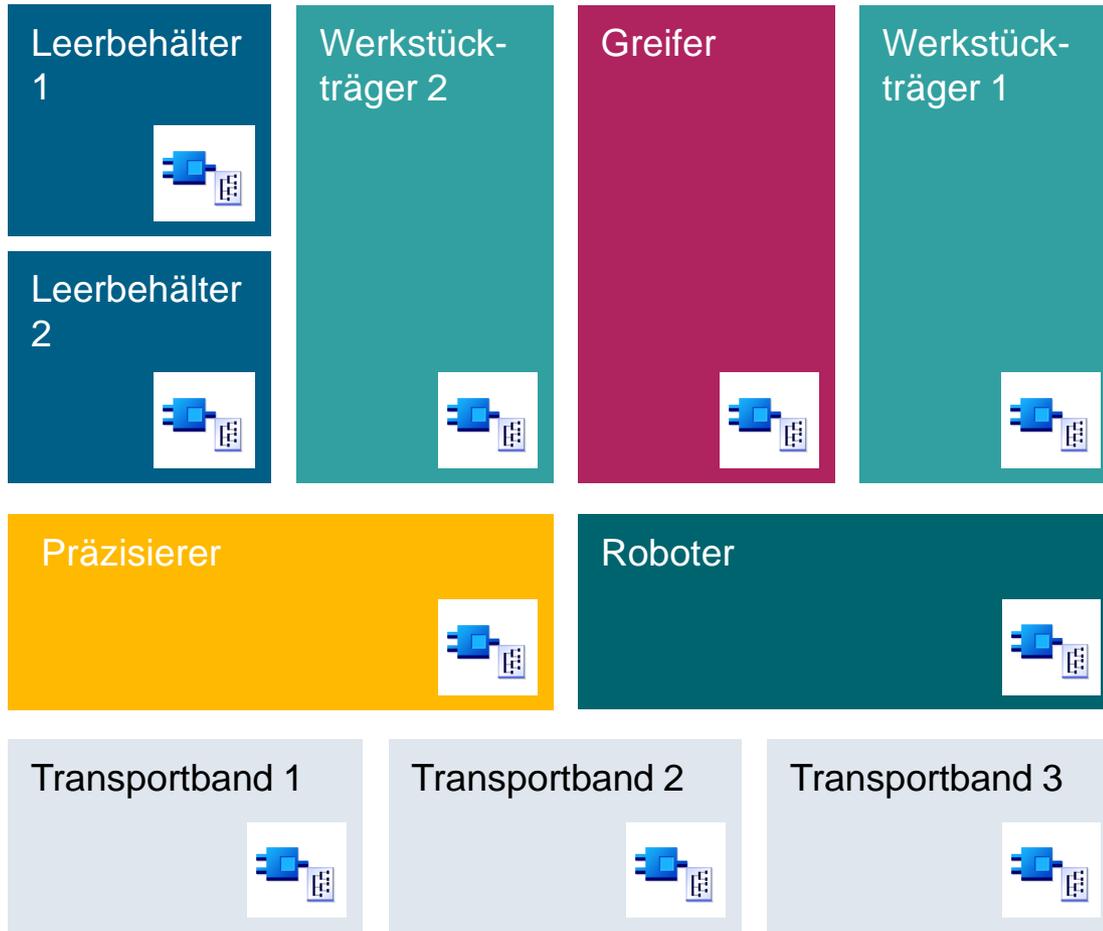


# TIA Digitalization Demonstrator – Übersicht

**SIEMENS**  
*Ingenuity for Life*



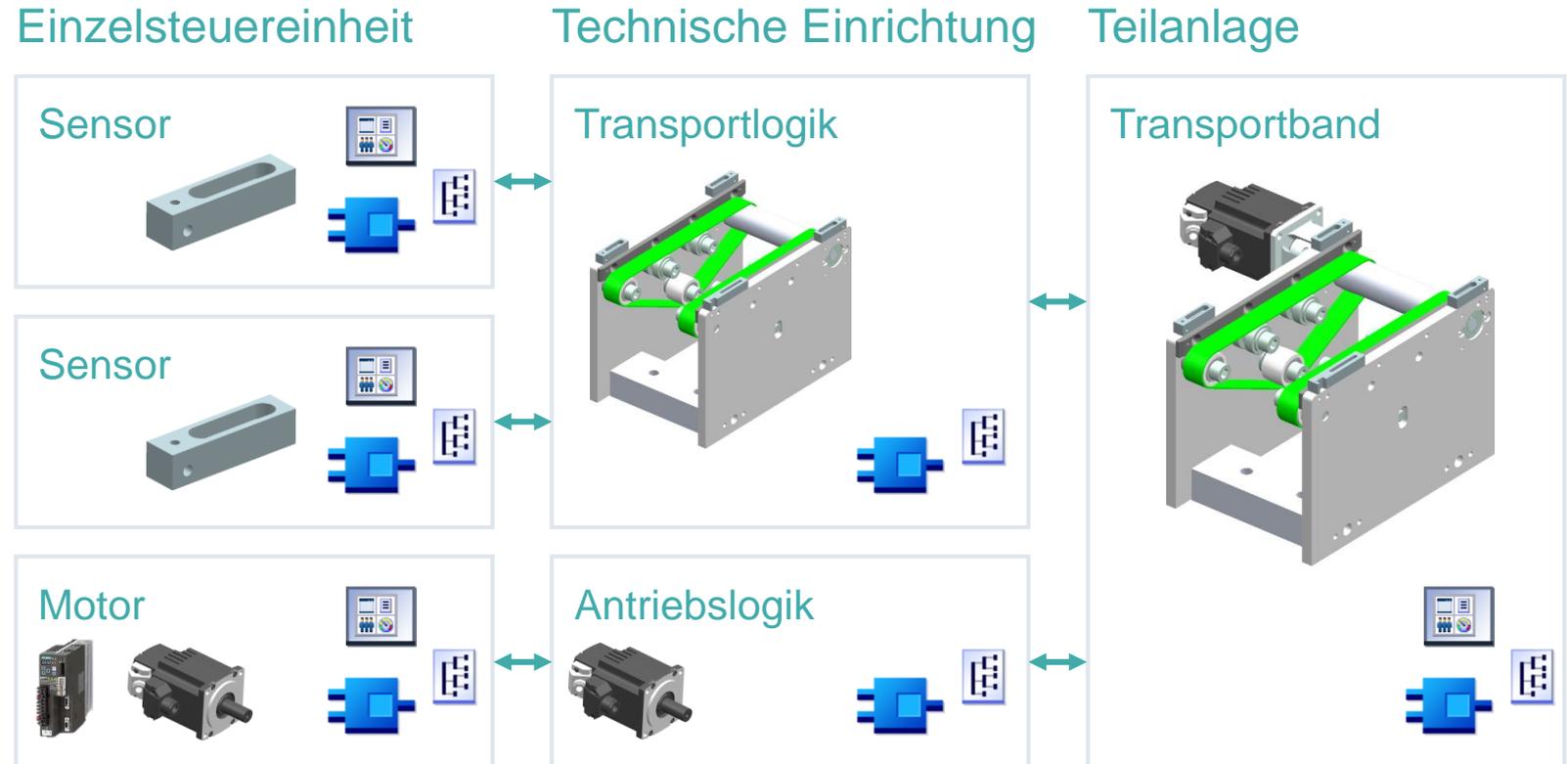
# TIA Digitalization Demonstrator – Funktionaler Schnitt



# TIA Digitalization Demonstrator – Verschalten von Objekten zu Funktionseinheiten

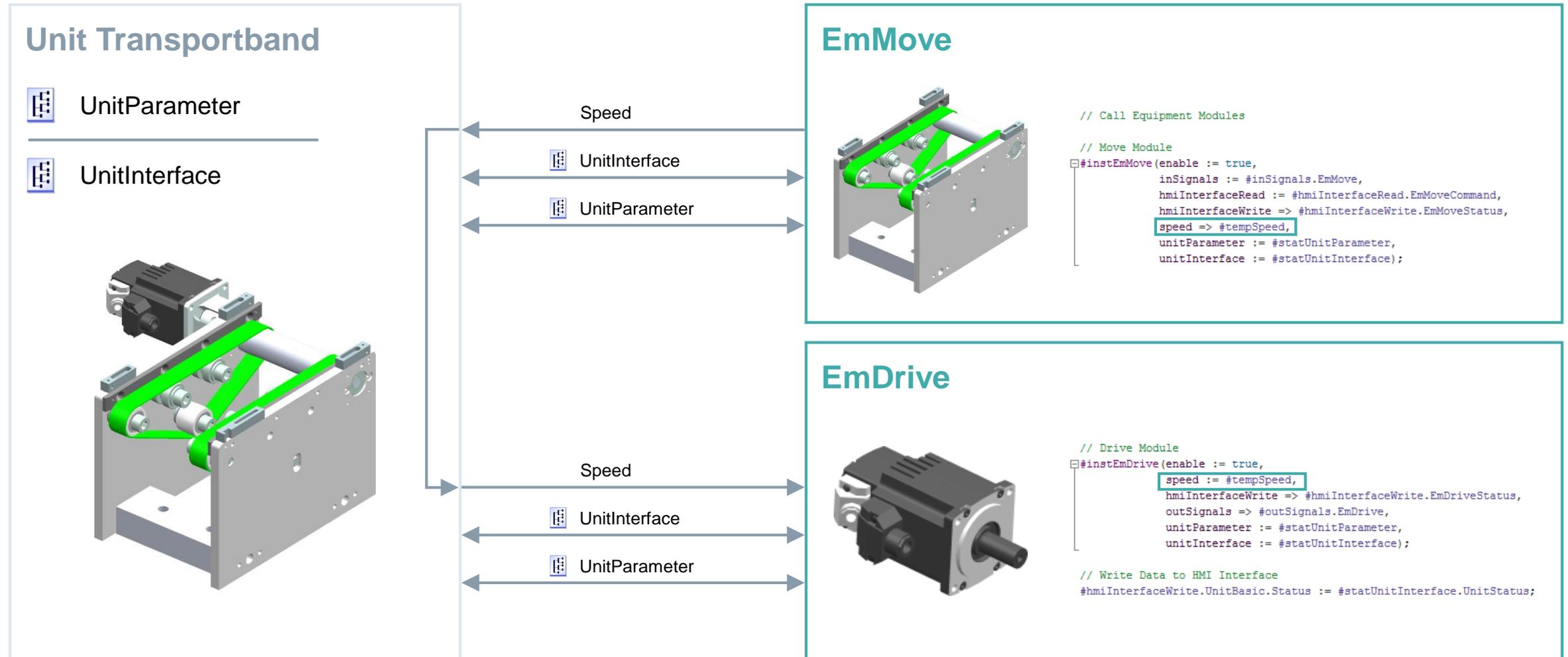
## Modularisierung

- Verschalten der technischen Einheiten zu logischen Einheiten und diese in Teilanlagen
- Jede Einheit kümmert sich um ihre Funktion
- Modularität bei Anpassungen (z. B. Austausch des Motors)



**Wie kann eine Standardisierung der Maschine vorgenommen werden?  
→ Durch sinnvolles Verschalten zu Funktionseinheiten**

# TIA Digitalization Demonstrator – Schnittstellendefinition



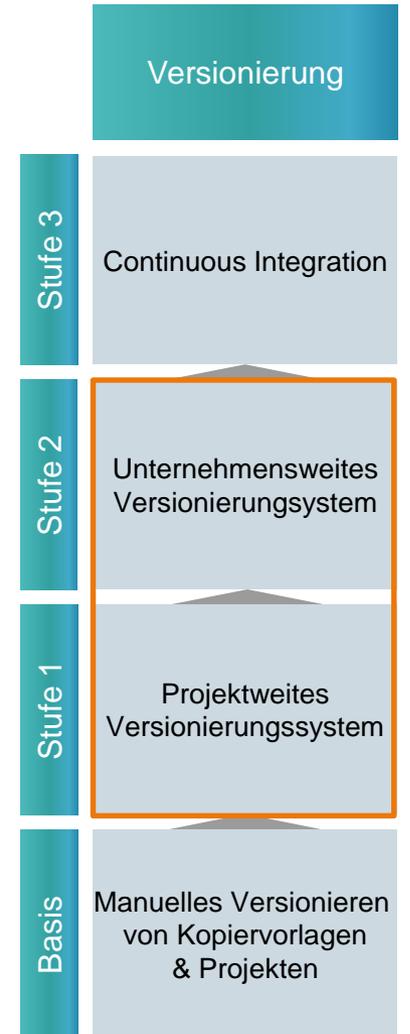
# Standardisierung an Hand eines Beispiels



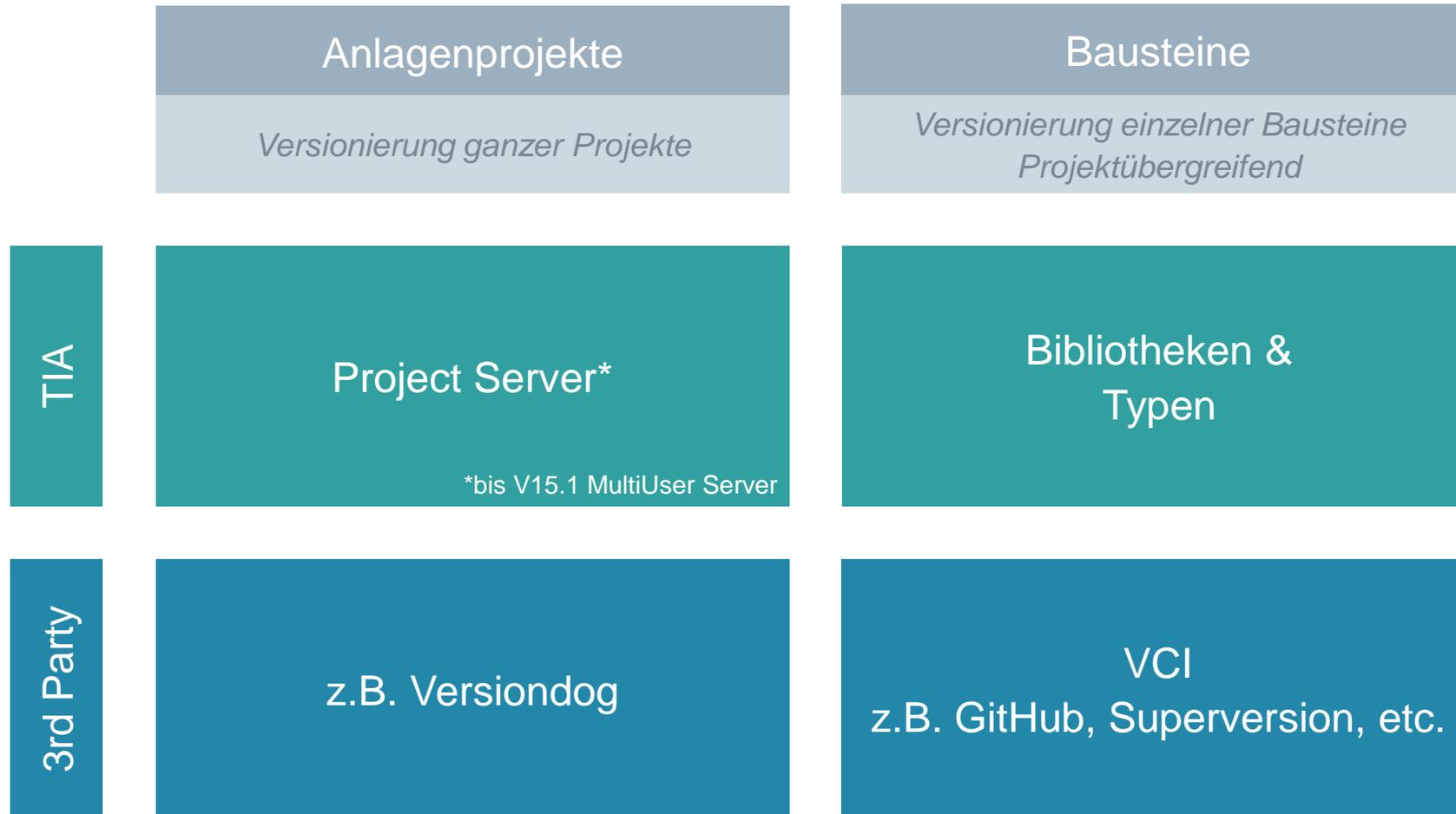
# Bibliothekskonzept

- Standardisierung mit TIA Portal

**SIEMENS**  
*Ingenuity for life*



# Übersicht Versionierung



# Übersicht Versionierung



# Bibliotheken im TIA Portal – Zwei Bibliotheksarten

## Projektbibliothek

- Im Projekt integriert
- Wird mit dem Projekt verwaltet (geöffnet, hochgerüstet, gespeichert, geschlossen)
- Ermöglicht die Wiederverwendung von Objekten innerhalb des Projekts
- Enthält Typen und Kopiervorlagen



## Globale Bibliotheken

- Eigenständige Bibliothek (unabhängig vom Projekt)
- Verwendung in mehreren Projekten möglich
- Zu Verteilung von fertigen Bibliotheksständen an andere Anwender
- Ablage z.B. auf einem Netzlaufwerk (Unternehmensbibliothek)
- Aktualisierungsaufforderung bei neuer Bibliotheksversion (Corporate Settings)
- Enthält Typen, Kopiervorlagen, „Gemeinsame Daten“ (Protokolle) und „Sprachen und Ressourcen“ (Bibliothekssprachen)

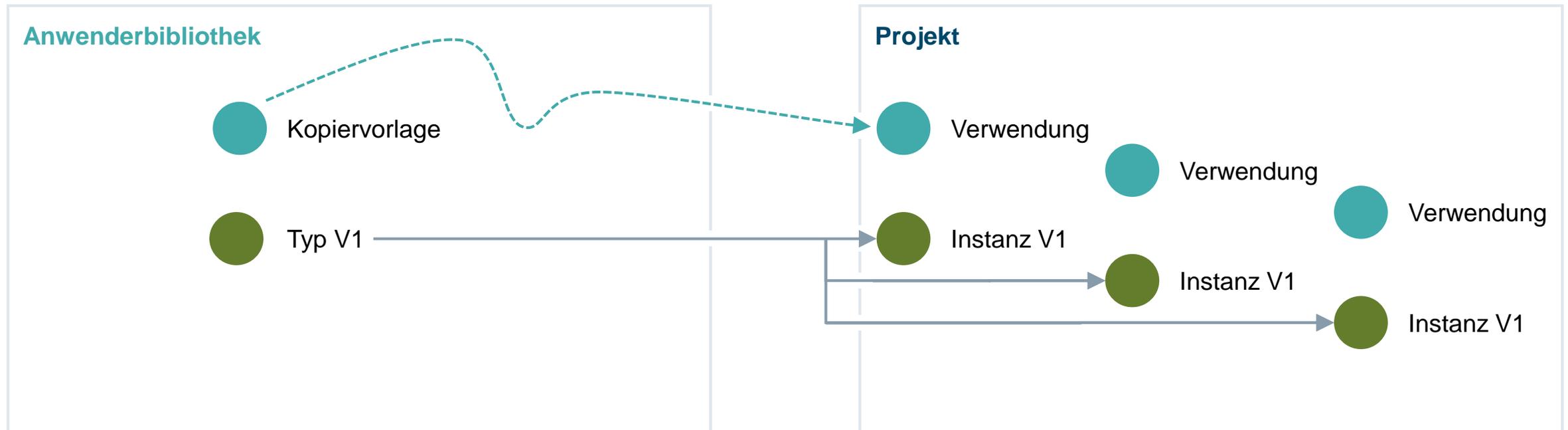
# Bibliothekskonzept – Wiederverwendbarkeit dank Kopiervorlagen und Typen

## Kopiervorlagen

- Einfache Ablage von Bausteinen, Hardwarekonfigurationen, HMI-Bilder, usw.
- Daten können auch inkonsistent abgelegt werden

## Typen

- Versionierung von Bausteinen
- Zentrale Updatefunktion aller Verwendungsstellen
- Typen sind schreibgeschützt
- Ablage nur für konsistente Daten (Kompilat wird mit abgelegt)



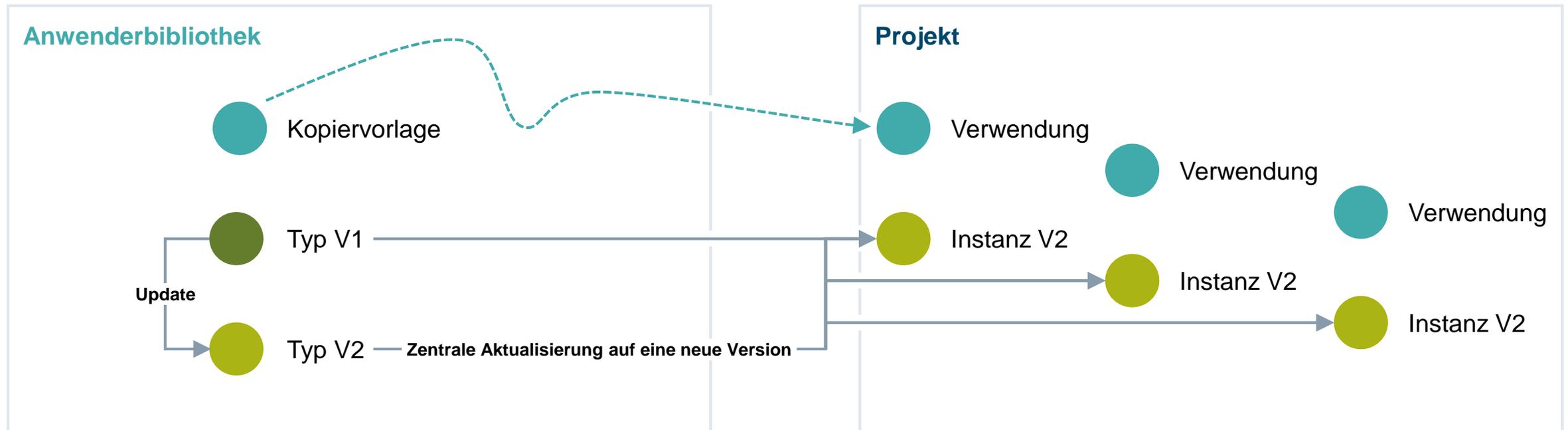
# Bibliothekskonzept – Wiederverwendbarkeit dank Kopiervorlagen und Typen

## Kopiervorlagen

- Einfache Ablage von Bausteinen, Hardwarekonfigurationen, HMI-Bilder, usw.
- Daten können auch inkonsistent abgelegt werden

## Typen

- Versionierung von Bausteinen
- Zentrale Updatefunktion aller Verwendungsstellen
- Typen sind schreibgeschützt
- Ablage nur für konsistente Daten (Kompilat wird mit abgelegt)



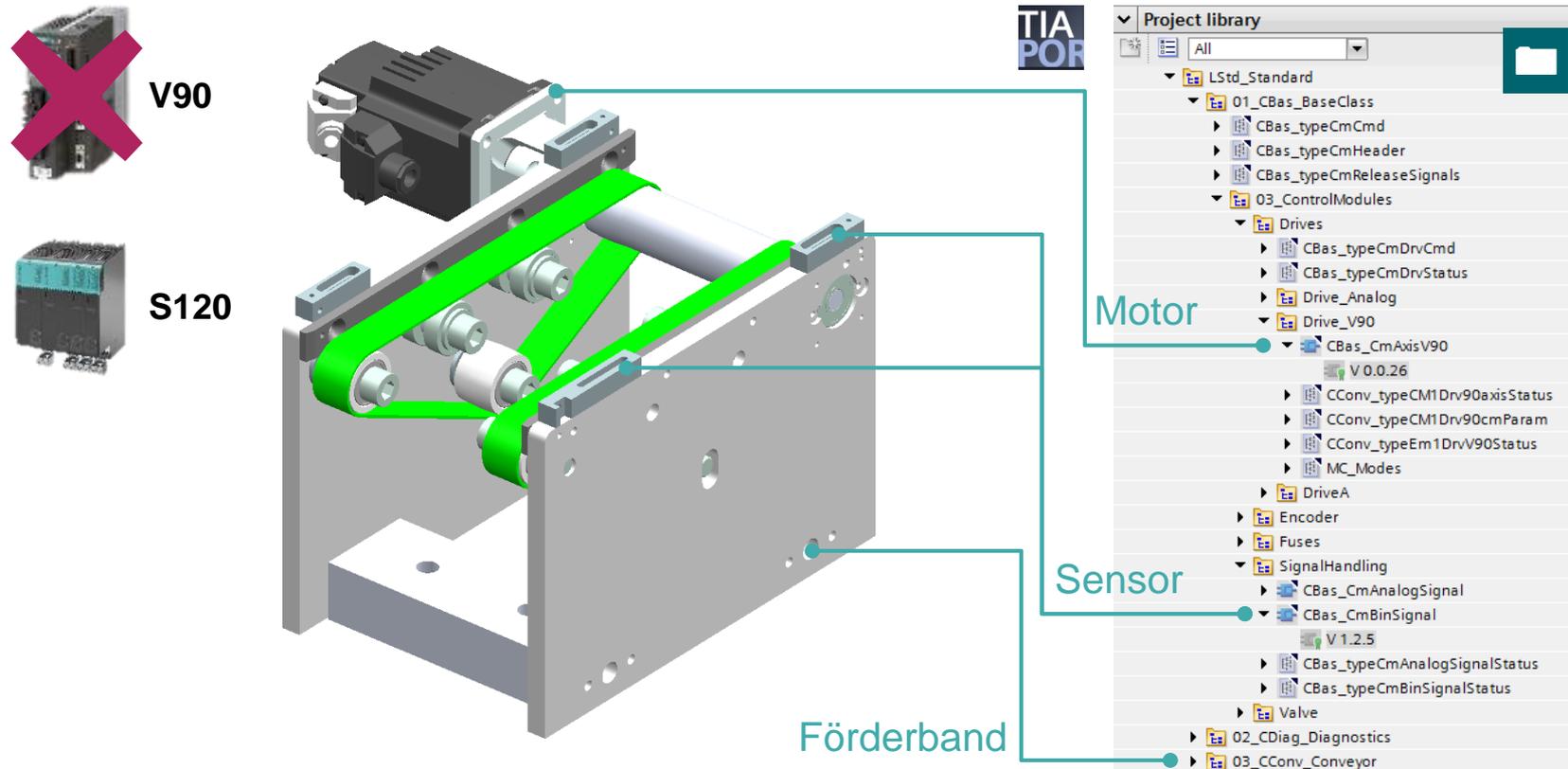
# Bibliothekskonzept – Übersicht Bibliothekselemente (PLC)

Bibliothekselement	Typ	Kopiervorlage
SIMATIC PLC		
OB	–	X
FB	X	X
FC	X	X
DB (Global)	–	X
Technologieobjekt	–	X
PLC-Variablen	–	X
PLC-Datentypen	X	X
Beobachtungs- und Forcetabellen	–	X
Traces	–	X
Textlisten	–	X

# Bibliothekskonzept – Übersicht Bibliothekselemente (HMI)

Bibliothekselement	Typ	Kopiervorlage
SIMATIC HMI		
Bilder	X	X
Bildbausteine (Faceplates)	X	–
Vorlagen (von Bildern)	–	X
Pop-Up-Bilder	–	X
Slide-in-Bilder	–	–
HMI-Variablen	–	X
Skripte	X	X
Protokolle	–	X
HMI-UDT	X	–
HMI Stil	X	–
HMI Formatvorlage	X	–
Benutzerverwaltung: Benutzer	–	X
Benutzerverwaltung: Gruppen	–	X

# Bibliothekskonzept – Verwaltung von Standards im TIA Portal



## Vorteile

- Wiederverwendbarkeit von validierten Objekten ermöglicht Zeitersparnis
- Minimierung der Testaufwände
- Nachverfolgbarkeit geänderter Objekte
- Qualitätssteigerung und Kosteneinsparung in der Softwareentwicklung
- Erhöhte Flexibilität durch einfache Anpassung von HW / SW Komponenten
- Automatisierung von Routingaufgaben

Objektverwaltung mit Hilfe des gesamtheitlichen Bibliothekskonzeptes

# Bibliothekskonzept - Übersicht

# Bibliotheksverwaltung – Übersicht im Projekt

Name	Typname	Version
LBFb_Actor	LBFb_Actor	V 0.0.1
LBFb_Clamp	LBFb_Clamp	V 0.0.1
LBFb_Conveyor	LBFb_Conveyor	V 0.0.1
LBFb_Drive	LBFb_Drive	V 0.0.1
LBFb_Glue	LBFb_Glue	V 0.0.1
LBFb_Heater	LBFb_Heater	V 0.0.1
LBFb_OperationMode	LBFb_OperationMode	V 0.0.1
LBFb_PlantGeneral	LBFb_PlantGeneral	V 0.0.1
LBFb_Press	LBFb_Press	V 0.0.1
LBFb_Sensor	LBFb_Sensor	V 0.0.1
LGF_Frequency	LGF_Frequency	V 1.1.5
LGF_Impulse	LGF_Impulse	V 1.2.2

Anzeige des Typen und der Version in der Projektnavigation

1. Tabellenkopf aktivieren
2. Mit Rechtsklick auf Tabellenkopf Spalten aktivieren

Typen

Projektbibliothek

Typ-Instanzen

Bibliotheksverwaltung öffnen, dazu die »Typen« selektieren

Bibliothekstexte exportieren/importieren

Globale Bibliothek

Symbol ohne Stift  
Bibliothek mit Leserechten geöffnet (Default)

# Übersicht Bibliotheksverwaltung – Verwendungsstellen und Versionen

## Bibliothek aufräumen

Unbenutzte Typen und Versionen entfernen

## Projekt harmonisieren

- Instanznamen im Projekt anpassen
- Pfade der Instanzen im Projekt anpassen
- Zentrale Verwaltung durch Typen möglich

## Filtermöglichkeiten

- In Arbeit
- Ohne Instanz
- Mehrere Versionen vorhanden
- Freigegeben
- Version Hochrüsten

Typ wird verwendet von...  
(überlagerte Typen)

The screenshot displays the 'Verwendungen im Projekt' (Usage in Project) window. It features two main tables:

Typen	Verwendet	Pfad
Basic Functions		
LBF_B_Actor		
V 0.0.1	UDT_Operating... UDT_HMIActor ...	Projektbibliothek\Typen\UDTs\UDT_OperatingMode\V 0.0.1 Projektbibliothek\Typen\UDTs\UDT_HMIActor\V 0.0.1
LBF_B_Clamp		
LBF_B_Conveyor		
LBF_B_Drive		
LBF_B_Glue		
LBF_B_Heater		
LBF_B_OperationMode		
LBF_B_PlantGeneral		
LBF_B_Press		
LBF_B_Sensor		

Typ-Versionen	Instanz im Projekt	Pfad
LBF_B_Actor		
V 0.0.1	LBF_B_Actor	ShoeProduction\PLC_1\Software Units\UnitStandard\Progr...

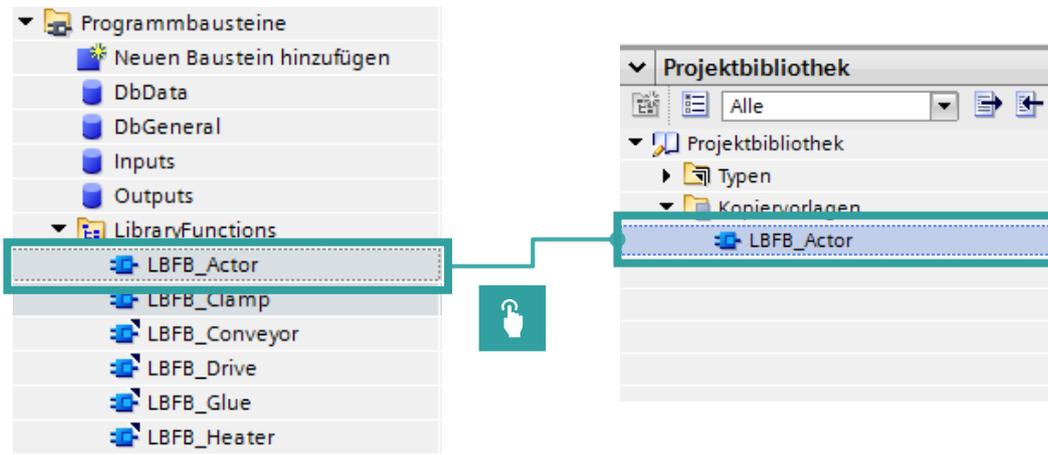
Annotations in the image:

- A box labeled 'Bibliothek aufräumen' points to the 'Basic Functions' and 'LBF\_B\_Actor' entries in the left table.
- A box labeled 'Projekt harmonisieren' points to the 'Verwendet' column in the left table.
- A box labeled 'Filtermöglichkeiten' points to the top toolbar of the interface.
- A box labeled 'Typ wird verwendet von... (überlagerte Typen)' points to the 'Instanz im Projekt' column in the right table.
- A box labeled 'Typ verwendet folgende Typen (unterlagerte Typen)' points to the 'Verwendet' column in the left table.

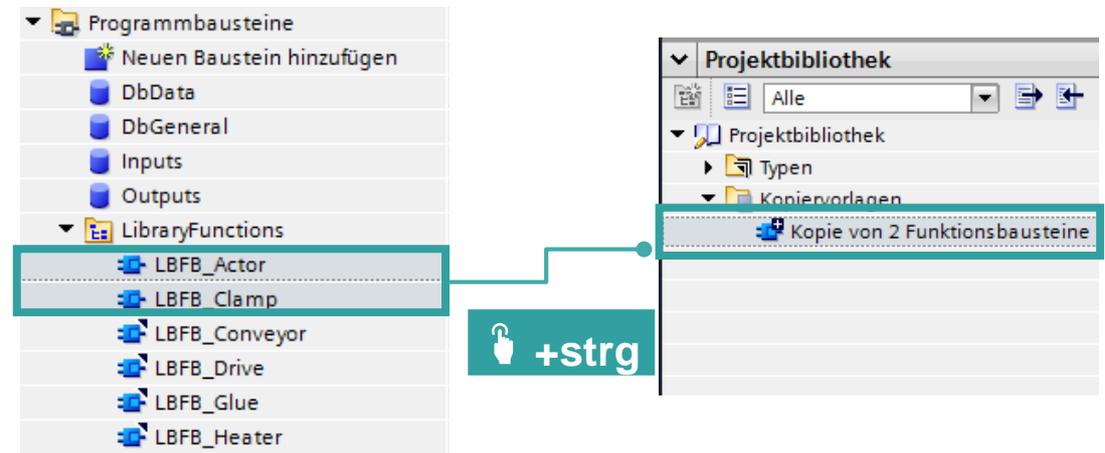
# Bibliothekskonzept - Kopiervorlagen

# Kopiervorlagen erstellen

Ablage von verschiedenen Objekten als  
Kopiervorlagen per drag & drop



Mehrere Bausteine können als Gruppe abgelegt  
werden



# Kopiervorlagen verwenden

Einfügen in das Gerät (PLC oder HMI)  
per drag & drop

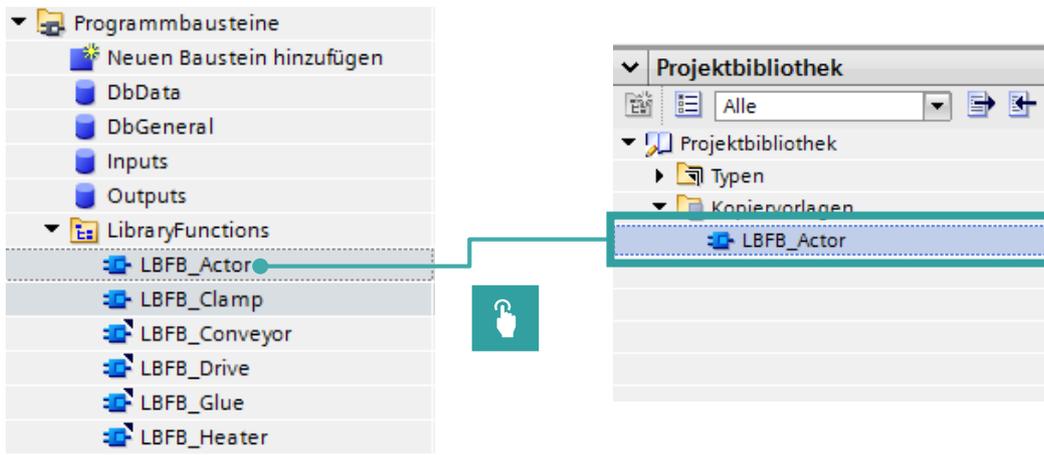
Unterstützt werden:

PLC:

OB, FB, FC, DB (global), TO, PLC-Variablen, PLC-Datentypen, Beobachtungs- und Forcetabellen, Traces, Textlisten

HMI:

Bilder, Vorlagen (von Bildern), Pop-Up-Bilder, HMI-Variablen, Skripte, Protokolle, Benutzerverwaltung: Benutzer und Gruppen



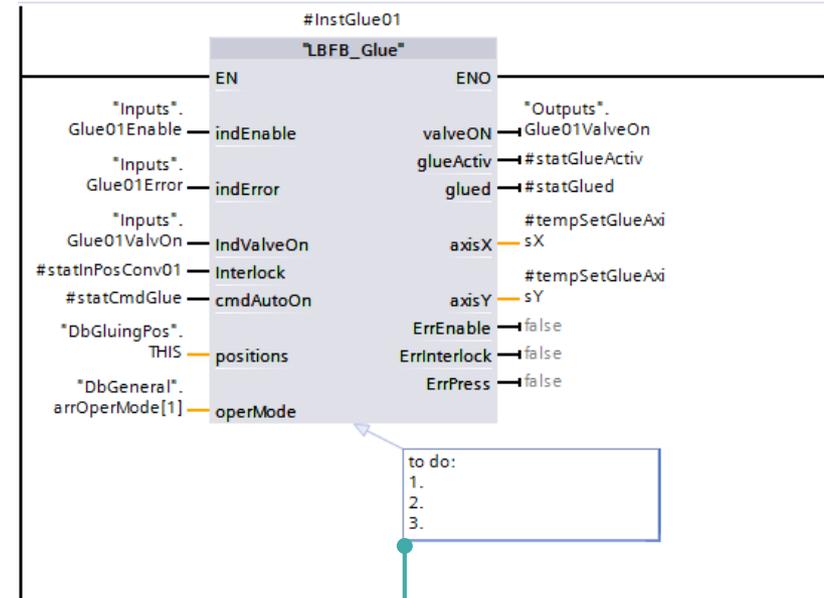
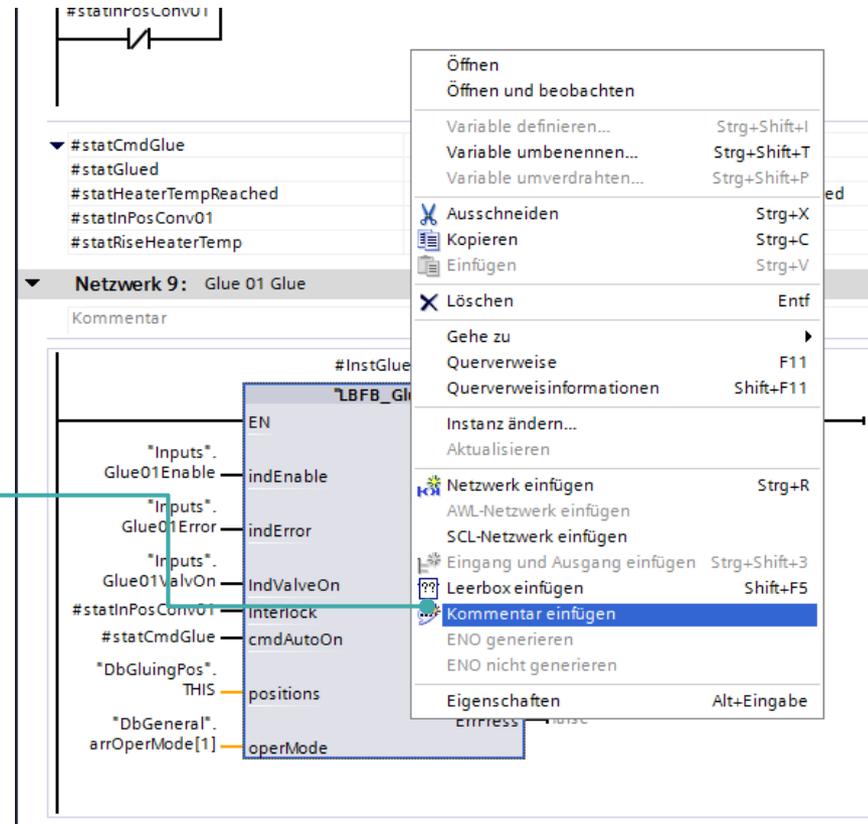
**X** Keine Versionierung, keine Verbindung zu Bibliothek!

# Freie Kommentare verwenden



**Freie Kommentare ein-/ausschalten**  
Blendet in graphischen Programmiersprachen die freien Kommentare ein/aus

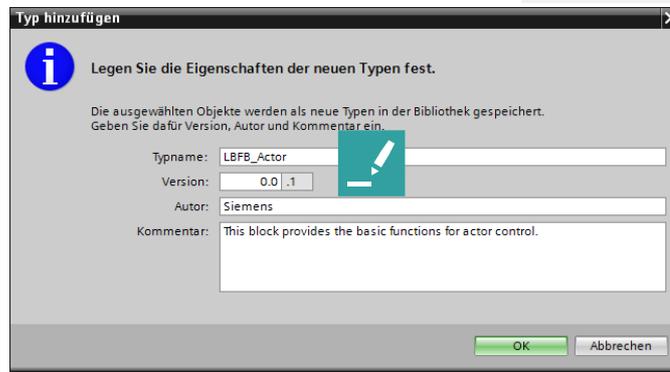
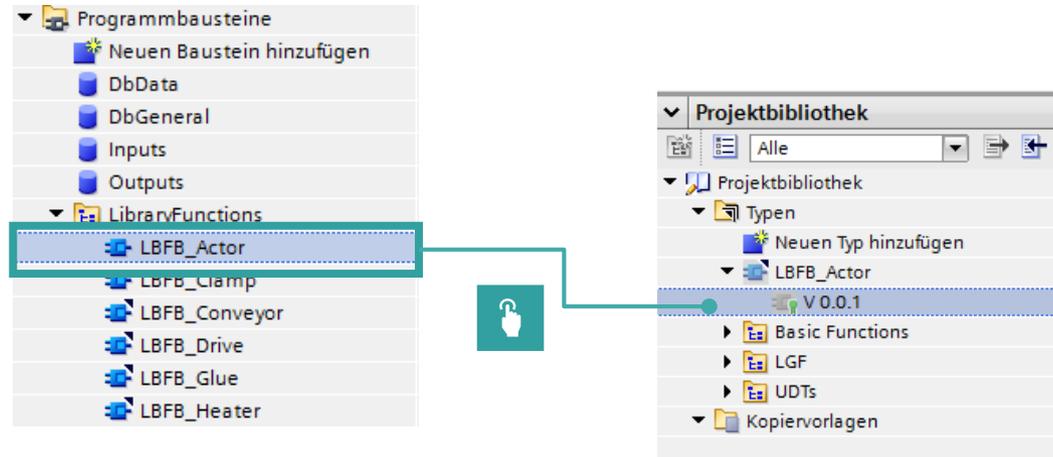
**Kommentar einfügen**  
Über den Rechtsklick auf eine Anweisung kann ein Kommentar hinzugefügt werden



**Freier Kommentar**  
Die Kommentarbox ist über einen Pfeil mit der zugehörigen Anweisung verbunden.  
Freie Kommentare werden mit der Kopiervorlage in der Bibliothek abgelegt.

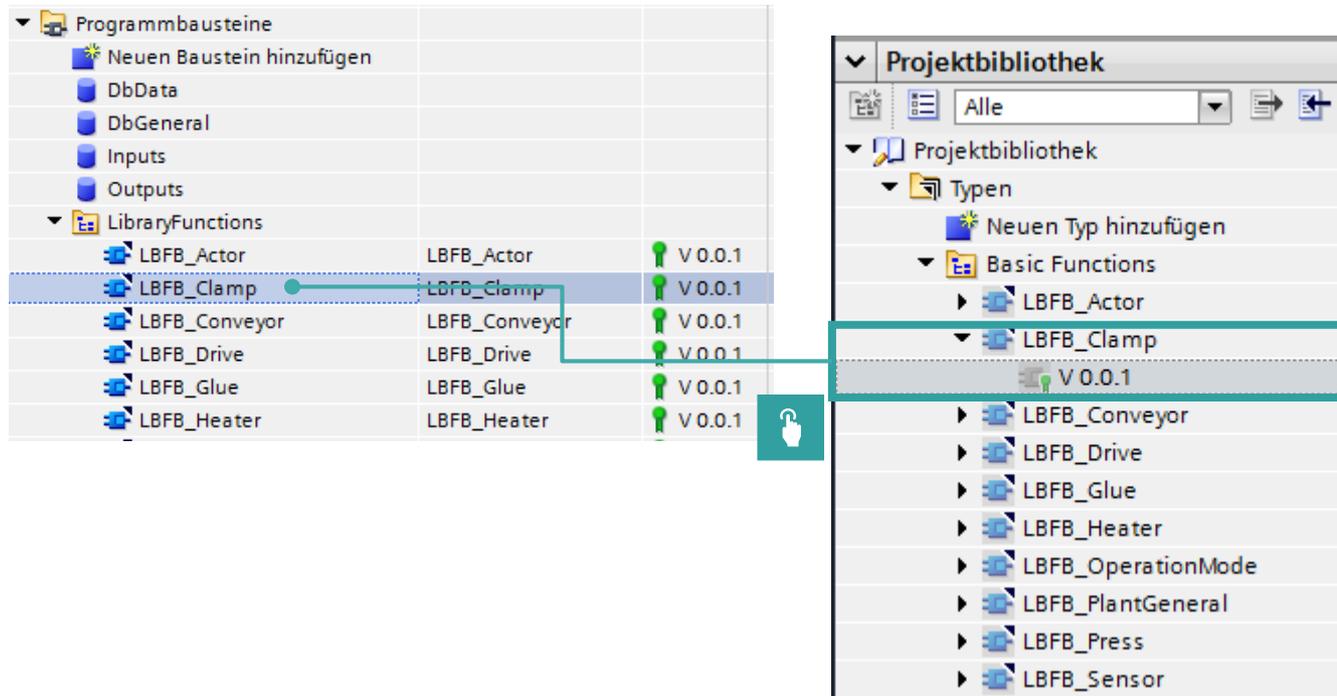
# Bibliothekskonzept - Typen

# Baustein als Typ in Projektbibliothek ablegen



- Typen sind versionierbar
- Erstellung eines Typs via drag & drop aus dem Gerät heraus
- Oder: (im Falle von manchen HMI-Objekten)
  - „Neuen Typ hinzufügen“ (in Projektbibl.)
- Ein Typ besitzt Metadaten
  - Name
  - Version
  - Autor
  - Kommentar

# Typisierte Bausteine in Projekt verwenden

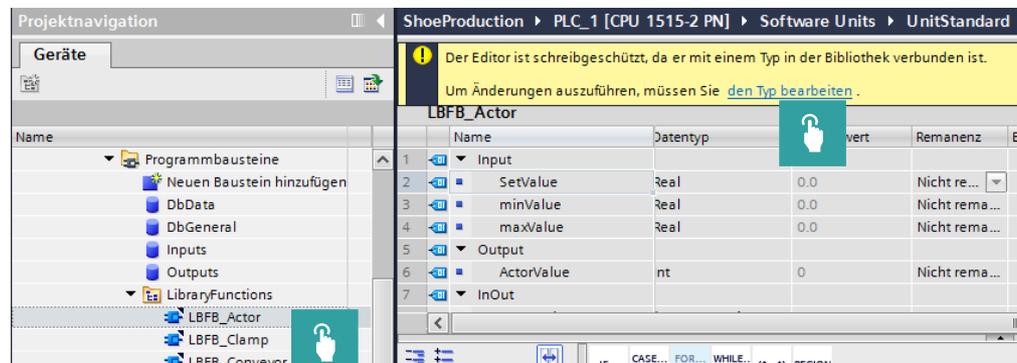


- Instanziierung der Typen via drag & drop
- Instanzen von Typen werden im Programm mit einem Dreieck im Objektsymbol, mit dem Typnamen und der Version gekennzeichnet (Spalten müssen ggfs. aktiviert werden)
- Unterstützt werden:
  - PLC: FB, FC, PLC-Datentypen
  - HMI: Bilder, Bildbausteine (Faceplates), Skripte, HMI UDT, Stile, HMI Formatvorlage

# Typen bearbeiten und freigeben

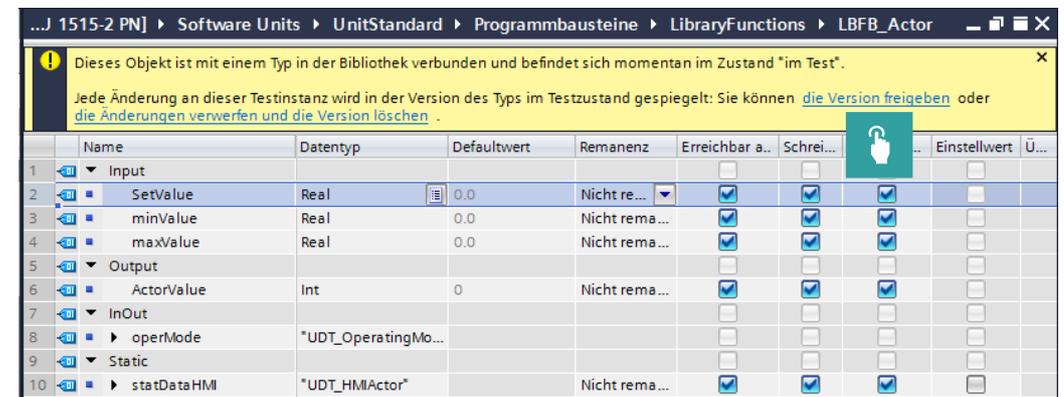
## Typ bearbeiten

- Doppelklick auf eine Typinstanz
- „den Typ bearbeiten“ (im gelben Fenster Oder
- In der Bibliotheksnavigation (rechte Seite) Rechtsklick auf den Typ
- „Typ bearbeiten“ anklicken
- Testumgebung festlegen



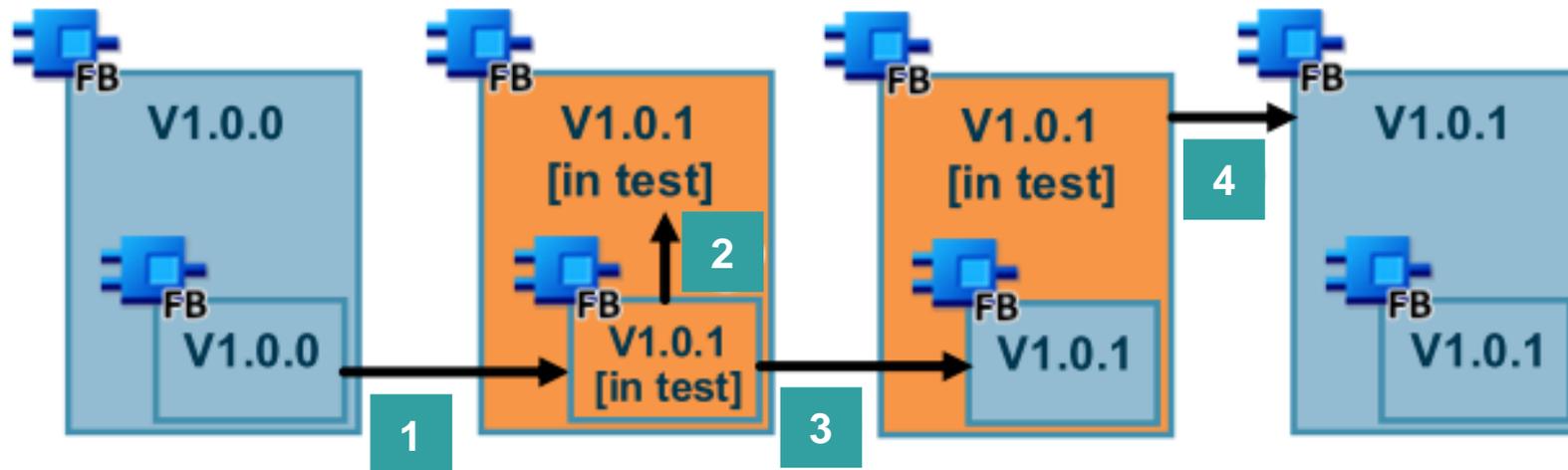
## Typ freigeben

- „diese Version freigeben“ -> alle Instanzen werden automatisch aktualisiert
- Eigenschaften festlegen
  - Typname
  - Version (letzte Stelle automatisch)
  - Autor
  - Kommentar
- Oder „die Änderungen verwerfen und die Version löschen“

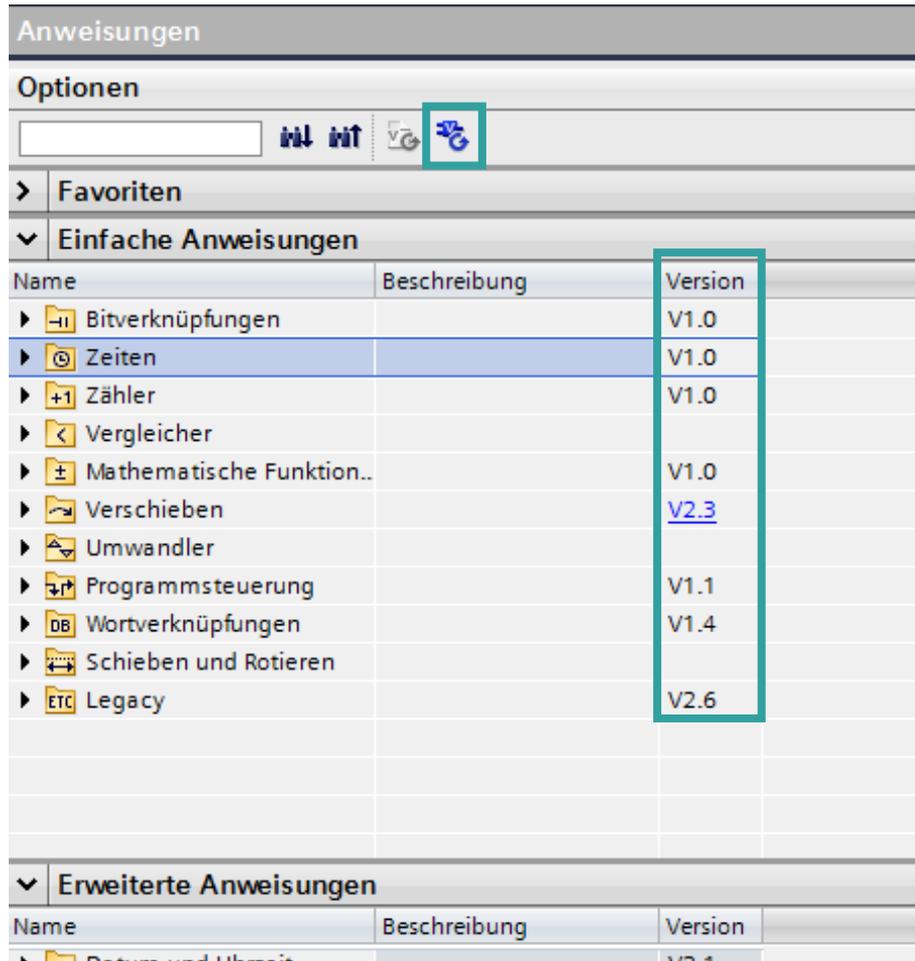


# Geschachtelte Typen bearbeiten und freigeben

1. Zu bearbeitenden Baustein [in Test] setzen
2. Überlagerter Typ wird automatisch auch in den Status [in Test] versetzt
3. Unterlagerten Baustein freigeben und neue Version vergeben
4. Überlagerten Baustein freigeben und neue Version vergeben



# Anweisungen mit Version



Name	Beschreibung	Version
▶ Bitverknüpfungen		V1.0
▶ Zeiten		V1.0
▶ Zähler		V1.0
▶ Vergleicher		
▶ Mathematische Funktion..		V1.0
▶ Verschieben		<u>V2.3</u>
▶ Umwandler		
▶ Programmsteuerung		V1.1
▶ Wortverknüpfungen		V1.4
▶ Schieben und Rotieren		
▶ ETC Legacy		V2.6

- Standard PLC-Anweisungen sind versioniert
- Nur eine Version einer Anweisung pro CPU erlaubt
- Können beim Hochrüsten der CPU zentral im Programm hochgerüstet werden (kein Automatismus!)

# Versionsnummern

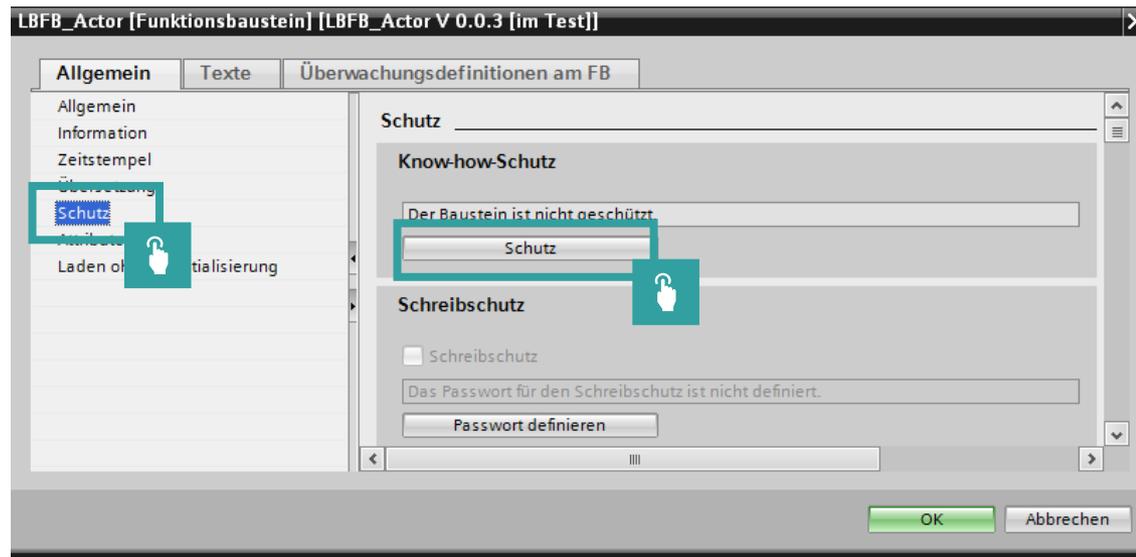
- Eine Version besteht immer aus drei Nummern (Vx.y.z)
- Zahlen von 1 bis 999 für die ersten beiden Stellen zulässig
- Die dritte Stelle wird automatisch inkrementiert
- Vorschlag zur Vergabe der Versionsnummern (aus dem Bibliotheksleitfaden):

Vx.y.z	Hauptrevisionsnummer	Nebenrevisionsnummer	Revisionsnummer
	<b>Nichtkompatible Änderung</b>	<b>Kompatible Änderung</b>	<b>Fehlerkorrektur</b>
	Reduzierung der Schnittstellen	Kompatible Schnittstellen	Fehlerkorrektur
	Ändern der vorhandenen Schnittstellen	Kompatible Erweiterung der Funktionalität	
	Inkompatible Erweiterung der Funktionalität		

# Know-How Schutz und Schreibschutz

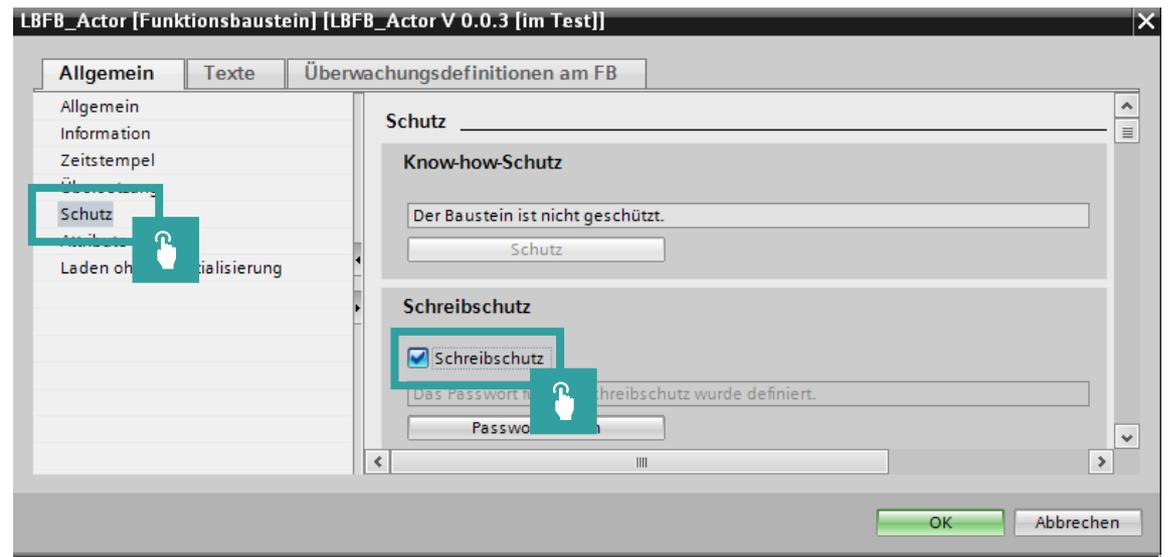
## Know-How Schutz

- Schützt die Bausteine vor unbefugtem Zugriff
- Der Know-How Schutz muss dem Baustein gegeben werden, bevor dieser zum Typen „ernannt“ wird
- Änderungen sind dann nur noch mit Passwort möglich



## Schreibschutz

- Schützt vor unbeabsichtigtem Ändern von Bausteinen
- Empfehlung: geschützte Bibliothek veröffentlichen



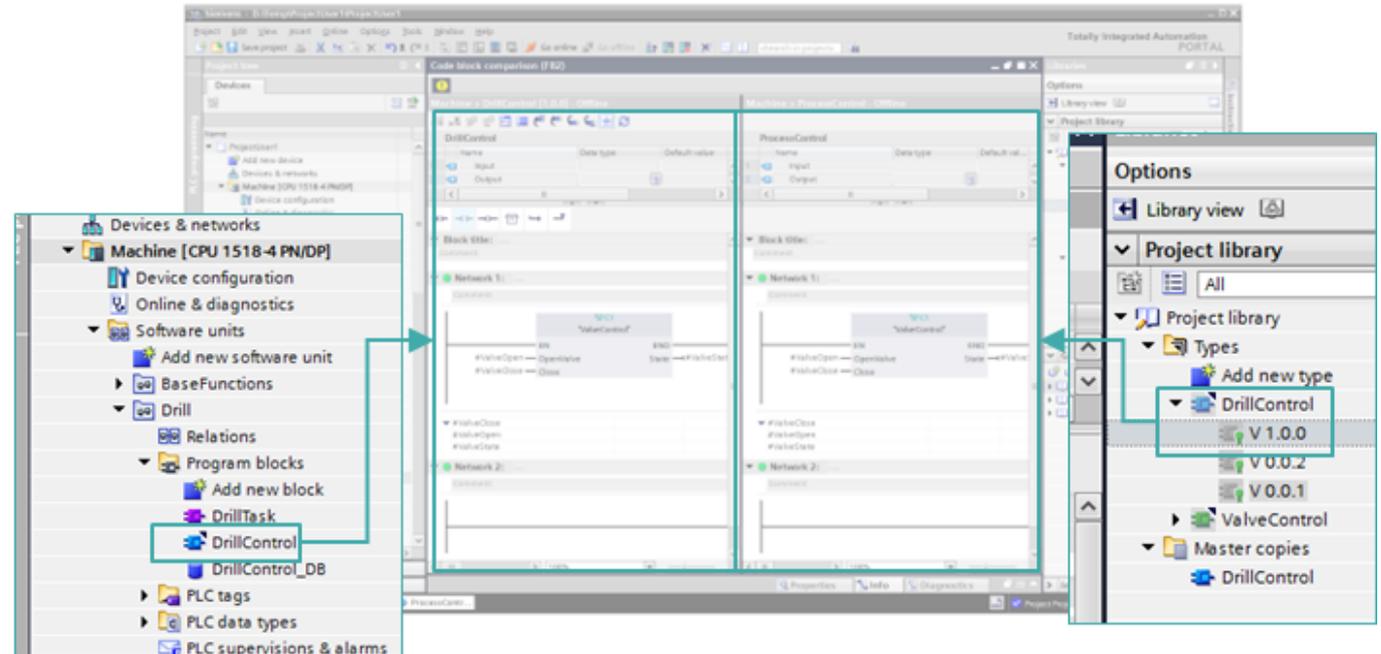
# Detaillierter Bausteinvergleich für Projekt <=> Bibliothek (neu ab V16)

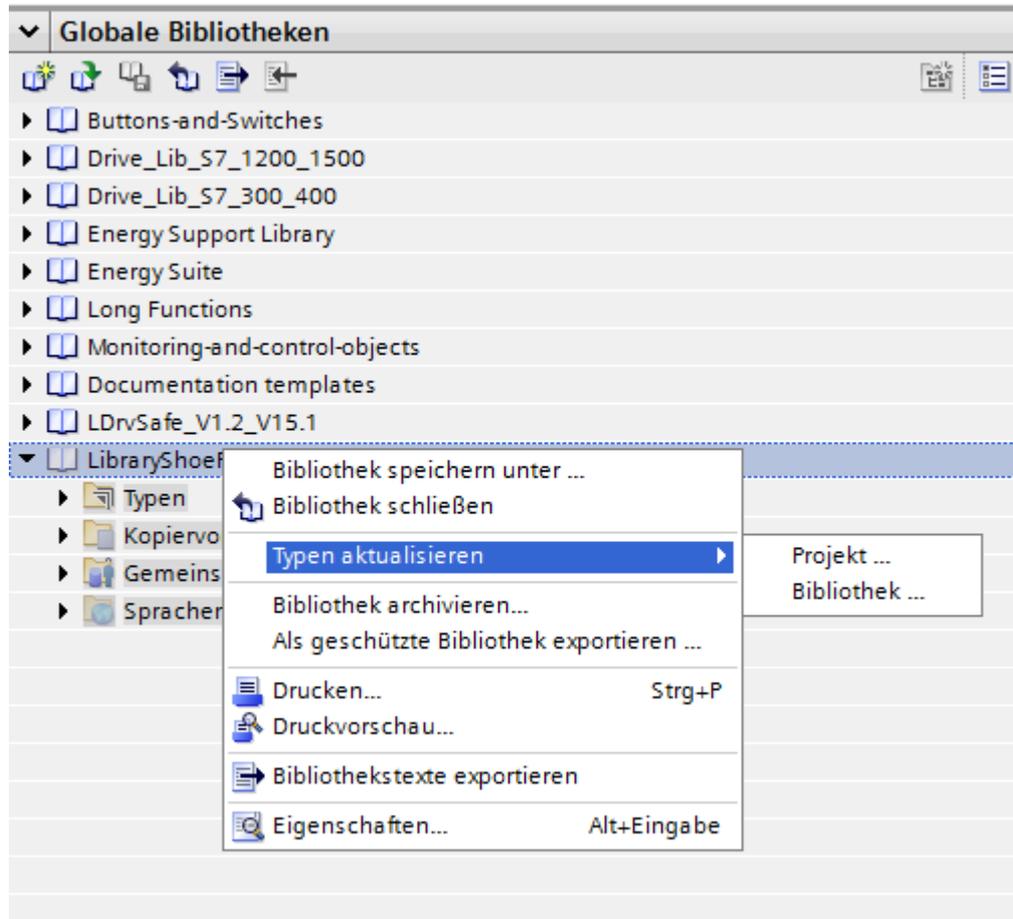
## Funktion

- Detaillierter Bausteinvergleich für Bausteine aus einem Projekt und
  - Masterkopien (Projekt oder globale Bibliothek)
  - Einzelne Versionen von Typen (aus Projekt oder globaler Bibliothek)
  - Über Schnellvergleich oder höheren Vergleichseditor
- Detaillierter Bausteinvergleich zwischen Bibliotheksbausteinen (z. B. V1.0 und V2.0)

## Vorteil

- Einfachere Handhabung von Bausteinen in Bibliotheken
- Bessere Nutzung zum Verfolgen von Änderungen zwischen Typversionen

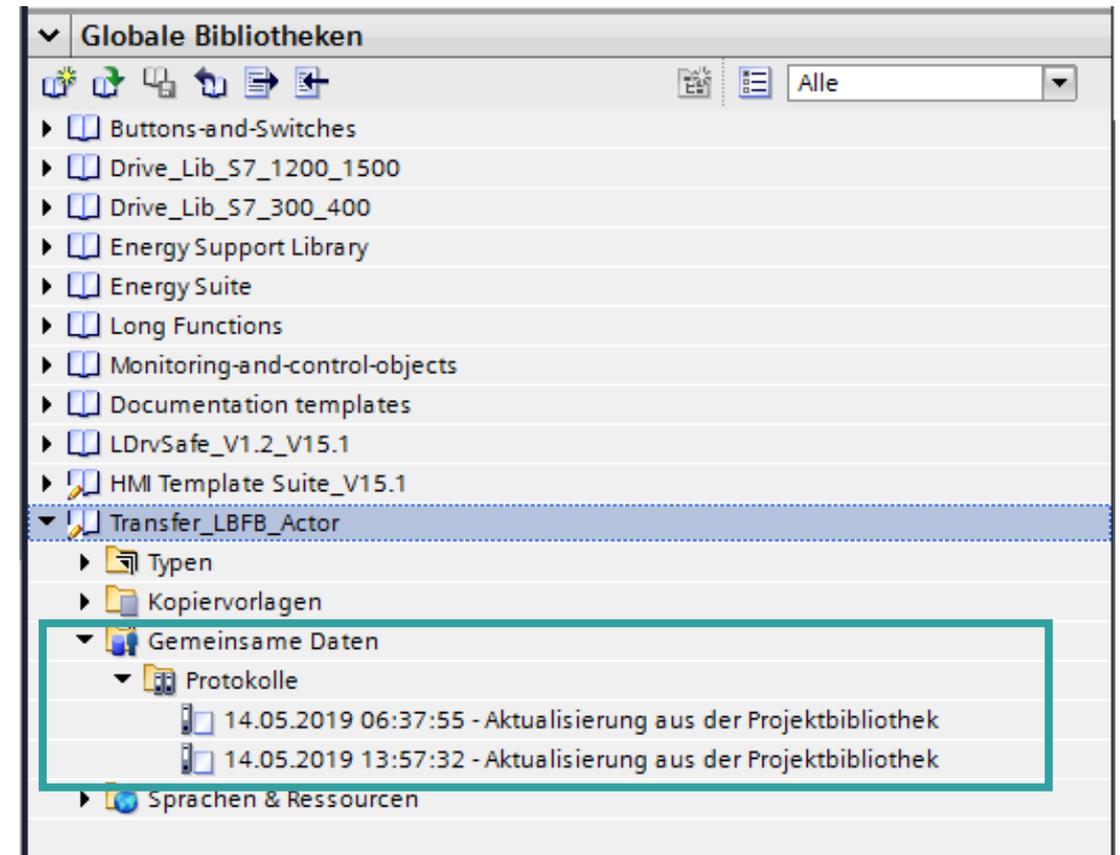
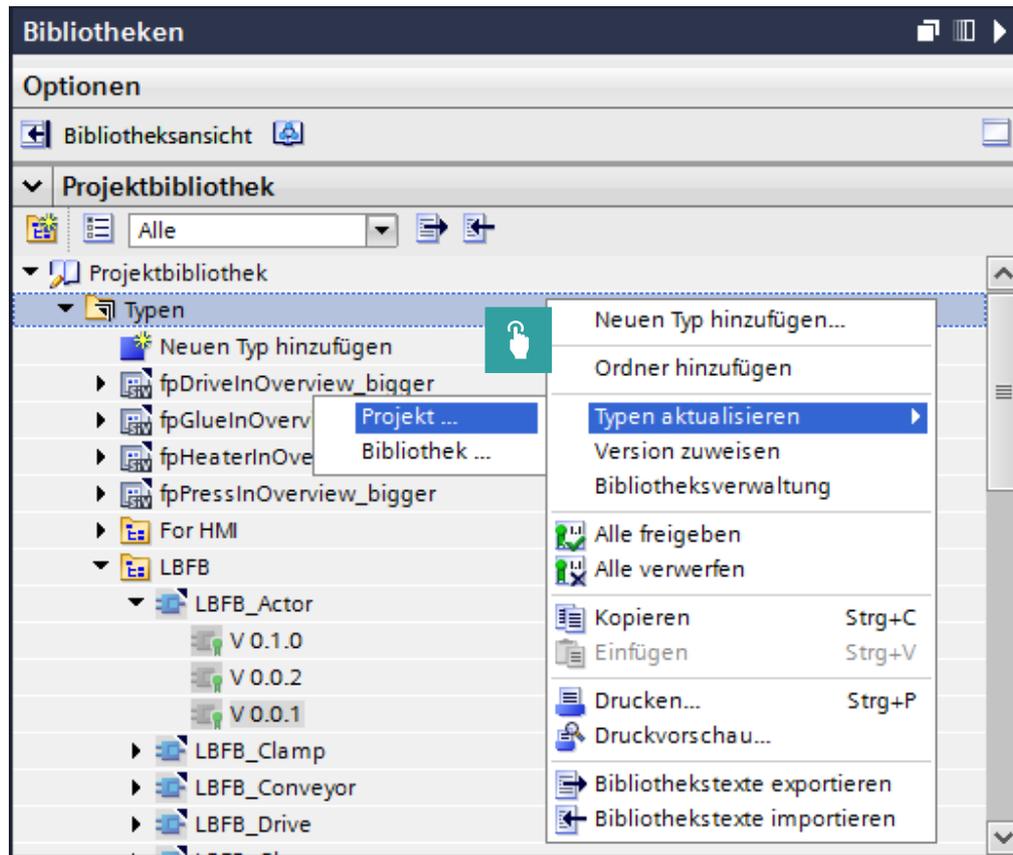




## Typen aktualisieren

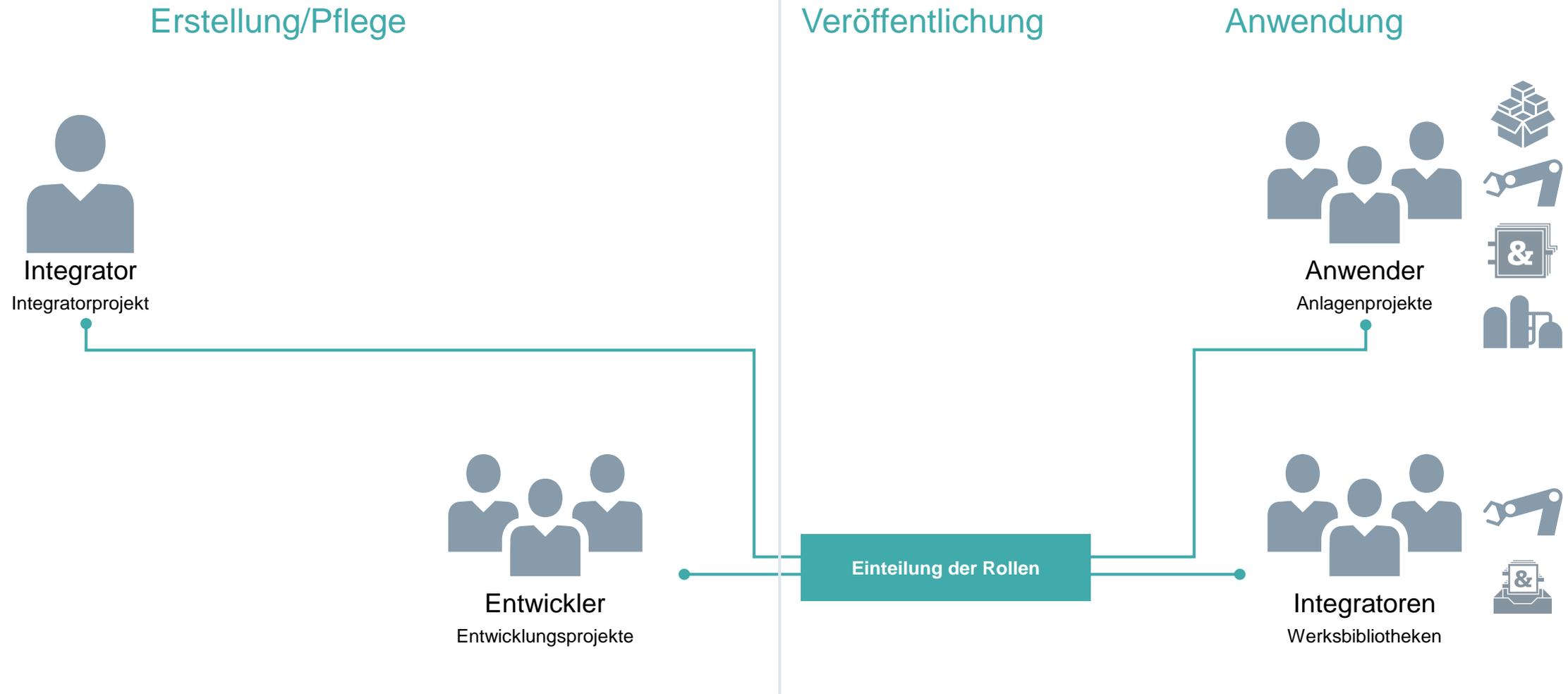
- Bidirektionale Aktualisierung von ...
  - Typen in Global- und Projektbibliothek
  - Instanzen im Projekt
- Die folgenden Elemente sind als Quelle für die Aktualisierung wählbar:
  - Gesamte Projektbibliothek
  - Einzelne Ordner
  - Einzelne (Mehrere) Typen
- Automatisches Protokoll des Aktualisierungsvorgangs wird generiert

# Bibliothek / Projekt hochrüsten



# Bibliotheks-konzept - Workflow

# Standardisierung im TIA Portal – Benutzerrollen beim Einsatz des Bibliothekskonzept



# Standardisierung im TIA Portal – Zugehörige Bibliotheken

Erstellung/Pflege

Veröffentlichung

Anwendung

Unternehmensbibliothek



**Integrator**  
Integratorprojekt



**Entwicklungsbibliothek**



**Freigabebibliothek**



**Anwender**  
Anlagenprojekte



**Transferbibliothek**  
Änderungen



**Entwickler**  
Entwicklungsprojekte



**Integratoren**  
Werksbibliotheken



# Standardisierung im TIA Portal – Prozess beim Arbeiten mit Bibliotheken



# Standardisierung im TIA Portal – Klare Regeln beim Arbeiten mit Bibliotheken



# Standardisierung mit Hilfe von TIA Portal Bibliotheken – Ein kontinuierlicher Engineering Workflow

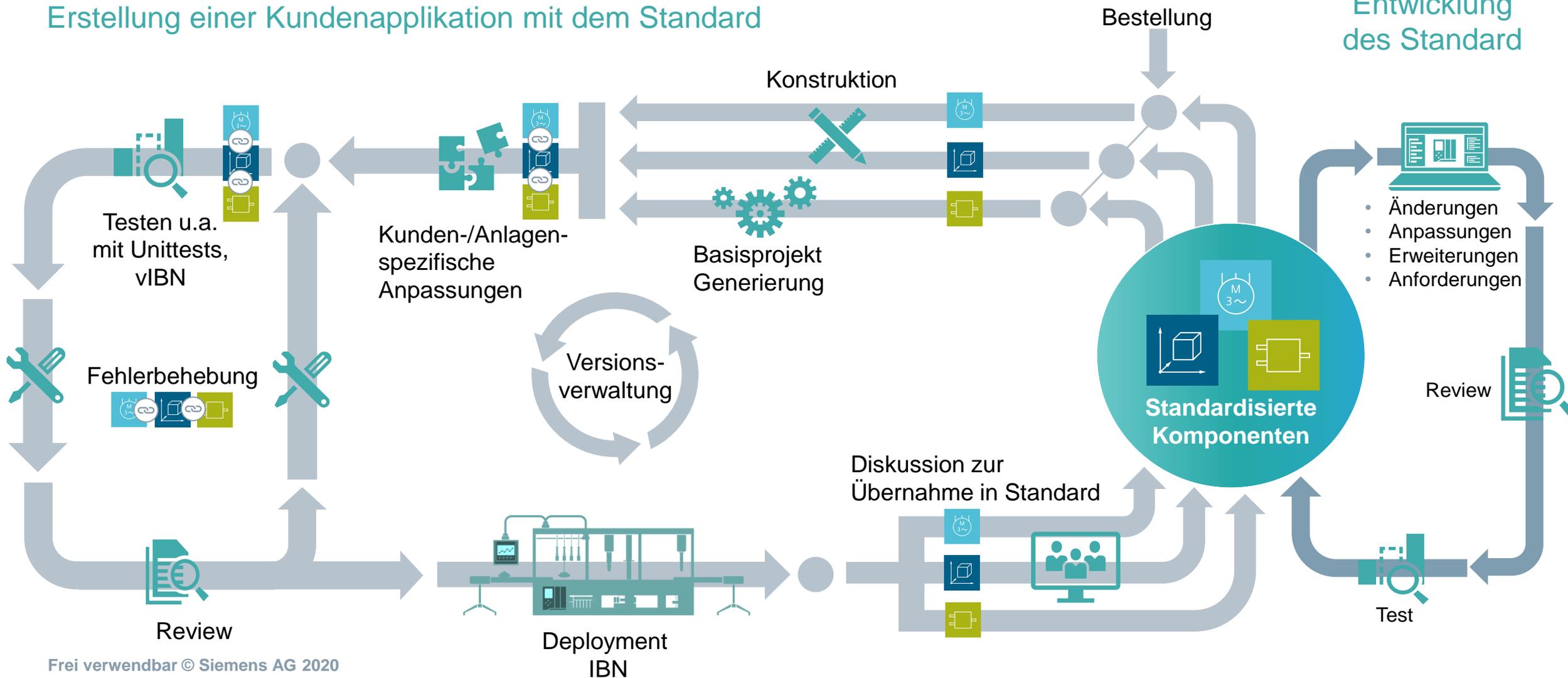
Erstellung einer Kundenapplikation mit dem Standard

Entwicklung des Standard



# Standardisierung für ein kontinuierlichen Engineering – Workflow

Erstellung einer Kundenapplikation mit dem Standard



# Bibliothekskonzept - Dokumentation

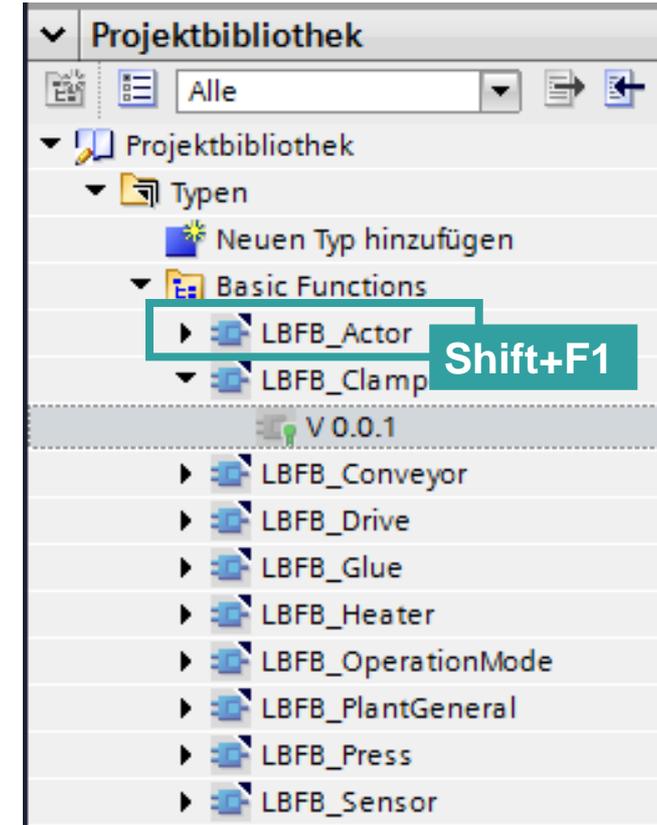
# Anwenderdefinierte Dokumentation verwenden

Aufrufbar über „Shift+F1“ für folgende Elemente des TIA Portals:

- Projekte /Bibliotheken
- Typen und Kopiervorlagen
- OBs, FBs, FCs und DBs
- Ordner
- Bilder (und Instanzen von Bildbausteinen)
- „Links“

## Unterstützte Dateiformate

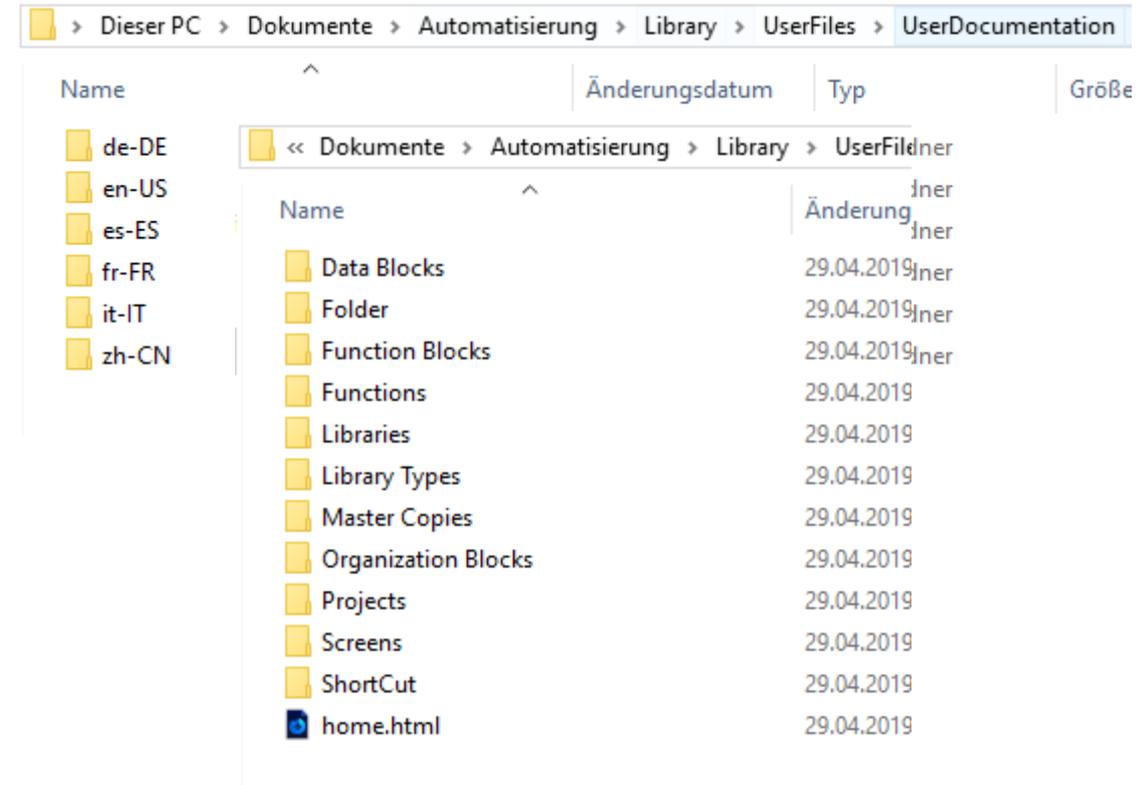
- Microsoft Word / Excel / Powerpoint (.docx / .xlsx / .pptx / .ppsx)
- HTML-Seiten & Compiled HTML (.htm / .html / .chm)
- Microsoft XPS (.xps)
- Rich Text Format (.rtf)
- Textdokumente (.txt)
- PDF-Dokumente (.pdf)



# Anwenderdefinierte Dokumentation verwenden – Verzeichnisstruktur anlegen

Ablage in einem der folgenden Verzeichnisse:

- Projektverzeichnis:  
*UserFiles\UserDocumentation\<Ordner für die jeweilige Sprache>\<Objektkategorie>*
- Verzeichnis einer globalen Bibliothek:  
*UserFiles\UserDocumentation\<Ordner für die jeweilige Sprache>\<Objektkategorie>*
- Zentrales Verzeichnis auf der Festplatte oder einem Netzlaufwerk:  
*<Zentrales Verzeichnis für anwenderdefinierte Dokumentation>\<Ordner für die jeweilige Sprache>\<Objektkategorie>*
- Beispiel:  
*C:\Users\UserName\Documents\Automatisierung\myProject\UserFiles\de-DE\Function Blocks\myFunction.pdf*



# Anwenderdefinierte Dokumentation verwenden

Zuordnung über Objektname (im TIA Portal) = Dateiname im Verzeichnis

The image shows two side-by-side screenshots from the Siemens TIA Portal. The left screenshot displays the 'Projektbibliothek' (Project Library) tree. Under 'Typen' (Types) > 'Basic Functions', the 'LFBF\_Actor' object is highlighted with a red box. The right screenshot shows a file explorer window with the path 'Automatisierung > Library > UserFiles > UserDocumentation > de-DE > Library Types'. A table lists the files in this directory, with 'LFBF\_Actor.html' highlighted by a red box. A red line connects the 'LFBF\_Actor' object in the library to the 'LFBF\_Actor.html' file in the file explorer, illustrating the mapping between the object name and the file name.

Name	Änderungsdatum	Typ	Größe
LFBF_Actor.html	10.05.2019 17:29	Firefox Document	0 KB

## Was sind die wesentlichen Vorteile?



- Software wird für den Anwender (Service, Instandhaltung) transparenter und somit leichter einsetzbar
- Qualität der Software wird erhöht und damit eine verbesserte Wartbarkeit im Service- und Diagnosefall sichergestellt
- Deutliche Reduzierung der Fehlerquellen durch die Verwendung bereits bewährter Programmteile (Module)
- Minimierung des Aufwands für Service und Instandhaltung
- Diagnose/Störungsbeseitigung wird vereinfacht
- Klare Dokumentation des Moduls mit festgelegtem Verhalten
- Reduktion von Entwicklungskosten

# Software Design / Bibliothekskonzept – Zusammenfassung

## Informationen



- Modularisierung von Programmcode nach DIN EN 61512-1
- Einheitlicher Programmierstil (Team-/ Firmenweit)
- Schnittstellendefinition
- TIA Portal Bibliotheken zur Versionierung von Programmcode
- Dokumentation durch Anwenderdefinierte Hilfe

## Unterlagen



- [Themenseite Standardisierung](#)
- [Leitfaden Standardisierung](#)
- [Programmierleitfaden und Styleguide](#)
- [Themenseite Bibliotheken im TIA Portal](#)
- [Leitfaden zur Bibliothekshandhabung im TIA Portal](#)

## Anwendungsfälle



- Verbesserung der Softwarequalität
- Wiederverwendbarkeit von Programmcode
- Vorbereitung auf den digitalen Wandel
- Versionierung / Update von Programmcode

## SITRAIN Kurse



- [Totally Integrated Automation im Digital Enterprise - Einführung in die Standardisierung \(1 Tag\)](#)
- [Totally Integrated Automation im Digital Enterprise - Einführung Kompakt \(1 Tag\)](#)
- [SIMATIC System-Umsteigerkurs auf SIMATIC S7-1500 im TIA Portal \(5 Tage\)](#)
- [SIMATIC Programmieren 3 im TIA Portal \(5 Tage\)](#)

# HMI Design

- Standardisierung mit TIA Portal

**SIEMENS**  
*Ingenuity for life*



## Die Herausforderung – Wiederverwendbarkeit von Objekten

Gibt es eine Möglichkeit, einmalig  
konfigurierte Objektgruppen als  
Vorlage abzulegen?



Ja, in WinCC können Bildbausteine  
erstellt und zentral in der Bibliothek  
verwaltet werden.

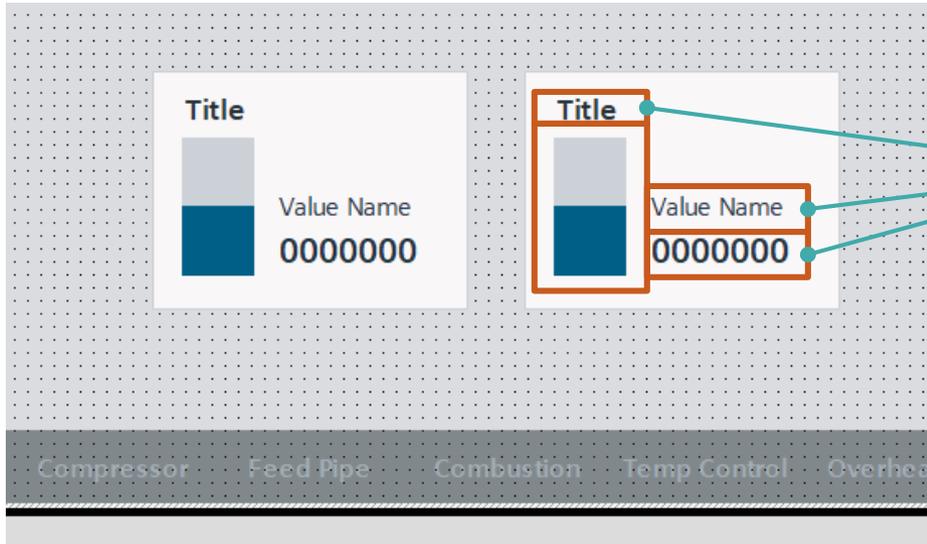


# HMI Design Bildbausteine (Faceplates)

- 1 **Was ist ein Bildbaustein?**
- 2 Wie werden Bildbausteine erstellt?
- 3 Welche Vorteile haben Bildbausteine?

# HMI Design – Bildbausteine

## Vergleich Objekte und Bildbaustein



Einzelne Objekte

Text field\_3 [Text field]

Bar\_2 [Bar]

VO field\_2 [VO field]

Properties Animations Events Texts

Property list

General

Appearance

Characteristics

Layout

Text format

Flashing

Limits

Styles/Designs

Miscellaneous

Security

Process

Tag:

PLC tag:

Address:

Type

Mode:

Style

Minimum scale value:

Font: Siemens San

Jedes Objekt hat seine eigenen Eigenschaften

# HMI Design – Bildbausteine

## Vergleich Objekte und Bildbaustein

The screenshot displays the Siemens HMI design environment. At the top, two identical widgets are shown on a grid background. The left widget is highlighted with a red border and a red box around it, with a red callout box pointing to it that says "Neuer Tab 'Schnittstelle'". The right widget is a standard object. A red callout box labeled "Bildbaustein" points to the right widget. Below the design area, the software's interface is visible, including a tab bar with "Properties", "Interface", "Animations", "Events", and "Texts". The "Properties" tab is active, showing a "Property list" on the left and a "Layout" section on the right. The "Layout" section includes "Position & size" fields for X (284), Y (211), Width (192), and Height (147), and a "Characteristics" field set to "None". A red callout box labeled "Keine einzelne Objekte, die konfiguriert werden können" points to the "Miscellaneous" section in the "Property list".

# HMI Design – Bildbausteine

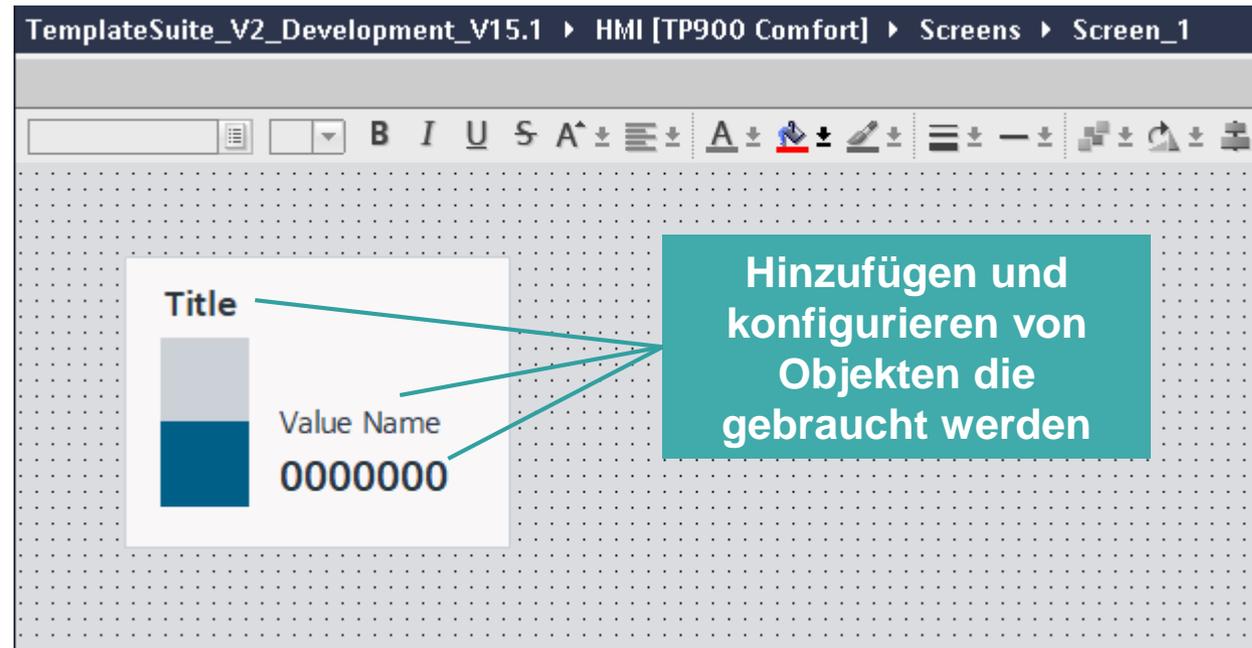
- 1 Was ist ein Bildbaustein?
- 2 **Wie werden Bildbausteine erstellt?**
- 3 Welche Vorteile haben Bildbausteine?

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

### Beispiel mit 4 Objekten

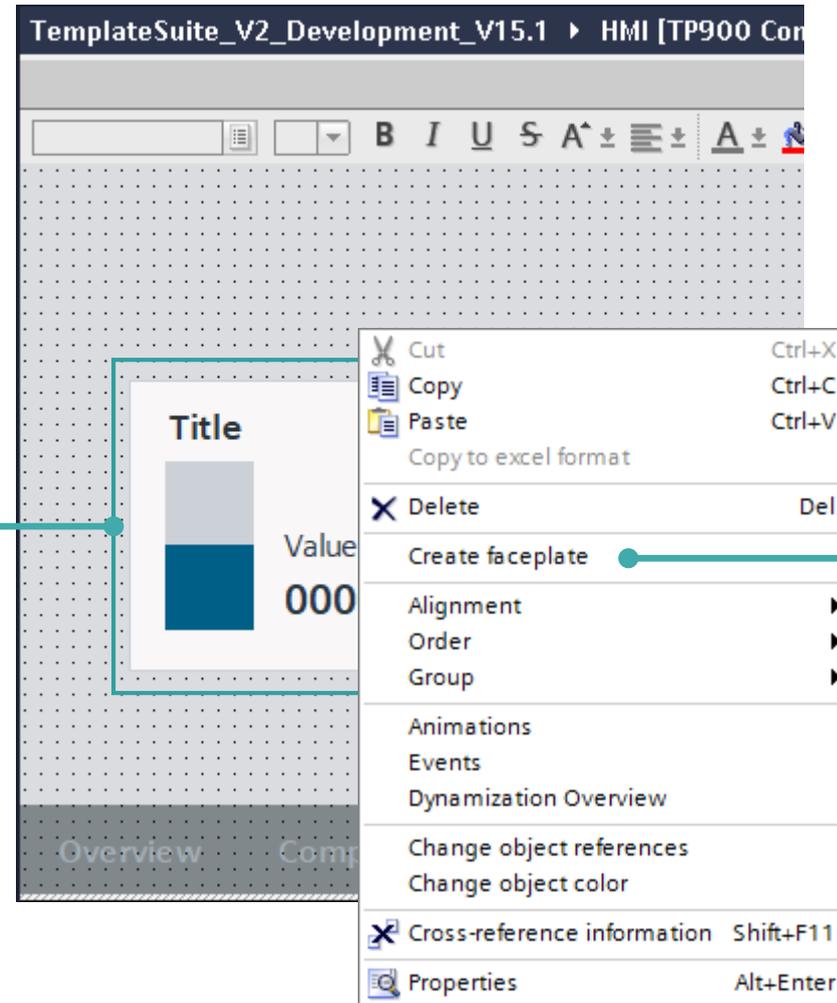
- Titel (Textfeld) enthält den Namen des Tanks welcher dargestellt wird
- E/A-Feld und Balken zeigen den gleichen Wert an



# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

Auswahl selektieren  
und Rechtsklick...



...und auf „Erzeuge  
Bildbaustein-Typ“  
klicken

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

**Add type** [X]

**i** Define the properties for the new types.

The selected objects will be stored as new types in the library.  
Specify the version, author and comment for this.

Name of type:

Version:

Author:

Comment:

Name, Autor und  
Kommentar vergeben

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

The screenshot shows the Siemens HMI Design software interface. The main workspace displays a design element with a yellow warning banner at the top: "This type version is currently in the 'in work' state. You can [release the version](#) or [discard the changes and delete the version](#)." A teal callout box labeled "Automatischer Wechsel in die Bibliothek" points to this banner. The left sidebar contains the "Project library" and "Global libraries" panels. A teal callout box labeled "Objekteditor" points to the "Contained objects" list, which includes "Bar\_2", "I/O field\_2", "Rectangle\_2", "Text field\_3", and "Text field\_4". A yellow callout box labeled "Configure a connection with drag-and-drop." points to the "Properties" panel, which shows a table with columns "Name" and "Type" under the "Properties\_Faceplate" section. The bottom panel shows the "Properties" tab for the selected object, with a teal callout box labeled "Eigenschaften" pointing to the "Fit to size" and "Size" settings. The right sidebar contains the "Toolbox" with sections for "Basic objects", "Elements", "Controls", and "My Controls". A teal callout box labeled "Schnittstellen-Editor" points to the "Interface" section of the toolbox.

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

The screenshot displays the Siemens HMI Design software interface. The top navigation bar includes tabs for Properties, Events, Tags, Scripts, Text lists, Graphic lists, Texts, and Languages. The main workspace is divided into three sections:

- Left Panel (Properties):** A tree view showing a container named 'Contain' with sub-elements: Bar\_2, I/O field\_2, Rectangle\_2, Text field\_3, and Text field\_4. A teal callout box labeled 'Schnittstellen-Editor' points to the 'I/O field\_2' element.
- Center Panel:** A large empty workspace. A yellow callout box contains the text 'Configure a connection with drag-and-drop.' with an arrow pointing towards the right panel.
- Right Panel (Properties):** A detailed view of the 'Properties\_Faceplate' object. A context menu is open, listing options: 'Add new category', 'Add new property', 'Edit', 'Copy to Excel format', 'Delete (Del)', and 'Rename (F2)'. A teal callout box labeled 'Neue Eigenschaft hinzufügen' points to the 'Add new property' option.

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

Drei Eigenschaften  
für Vier Objekte

The screenshot shows the Siemens HMI Design software interface. The 'Properties' tab is active, displaying a list of contained objects on the left and a table of properties for the selected 'Properties\_Faceplate' object on the right. A yellow callout box with the text 'Configure a connection with drag-and-drop.' is positioned between the two panels, with arrows pointing to the 'tagFilling' property in the table and the 'Contained objects' list. A teal callout box at the top center contains the text 'Drei Eigenschaften für Vier Objekte', with a line pointing to the 'tagFilling', 'txtTitle', and 'txtValueName' properties in the table.

Name	Type
▶ Bar_2	
▶ I/O field_2	
▶ Rectangle_2	
▶ Text field_3	
▶ Text field_4	
Interface	
Properties_Faceplate	
tagFilling	Int
txtTitle	WString
txtValueName	WString

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

**Prozesswert auswählen und via Drag & Drop mit der entsprechenden Eigenschaft verschalten**

Configure a connection with drag-and-drop.

Name	Type
Properties_Faceplate	
tagFilling	Int
txtTitle	WString
txtValueName	WString

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

The screenshot displays the Siemens HMI Design software interface. At the top, there are tabs for Properties, Events, Tags, Scripts, Text lists, Graphic lists, Texts, and Languages. The main workspace is divided into three sections:

- Contained objects:** A list on the left containing Bar\_2, I/O field\_2, Rectangle\_2, Text field\_3, and Text field\_4. A red dot is placed next to I/O field\_2.
- Diagram:** A central workspace showing a graphical representation of the objects. Arrows indicate connections from I/O field\_2 to the tagFilling property and from Text field\_3 to the txtValueName property.
- Interface:** A table on the right showing the properties of the selected object (Properties\_Faceplate):

Name	Type
tagFilling	Double
txtTitle	String
txtValueName	String

Two callout boxes provide additional information:

- A red box in the center states: "E/A-Feld und Balken werden auf die gleiche Eigenschaft verschaltet" (E/A field and bar are switched to the same property).
- A red box at the bottom states: "Textfelder werden mit den beiden Strings verbunden" (Text fields are connected to both strings).

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

The screenshot displays the 'Libraries' pane in Siemens HMI Design. The 'Project library' is expanded to 'Faceplate\_1', where version 'V 0.0.2 [in work]' is selected. A context menu is open over this version, listing options such as 'Open', 'Edit type', 'Duplicate type', 'Check consistency', 'Release version', 'Discard changes and delete version', 'Cut', 'Copy', 'Paste', 'Delete', 'Cross-reference information', 'Print...', 'Print preview...', and 'Properties...'. A yellow warning message is visible at the top right: 'This type version is currently in the "in work" state. You can [release the version](#) or [discard the changes](#).' A teal callout box on the left points to the selected version with the text 'Rechtsklick'. Another teal callout box on the right points to the 'Release version' menu item with the text 'Version freigeben'.

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

**Release type version**

**i** Define the properties for the released type version.

A new version will be released for the selected types.  
Assign them common properties or confirm the recommended properties.

Name of type: Faceplate\_1

Version: 0.0 .2

Author: siemens

Comment:

**Options**

- Update instances in the project
- Delete unused type versions from the library

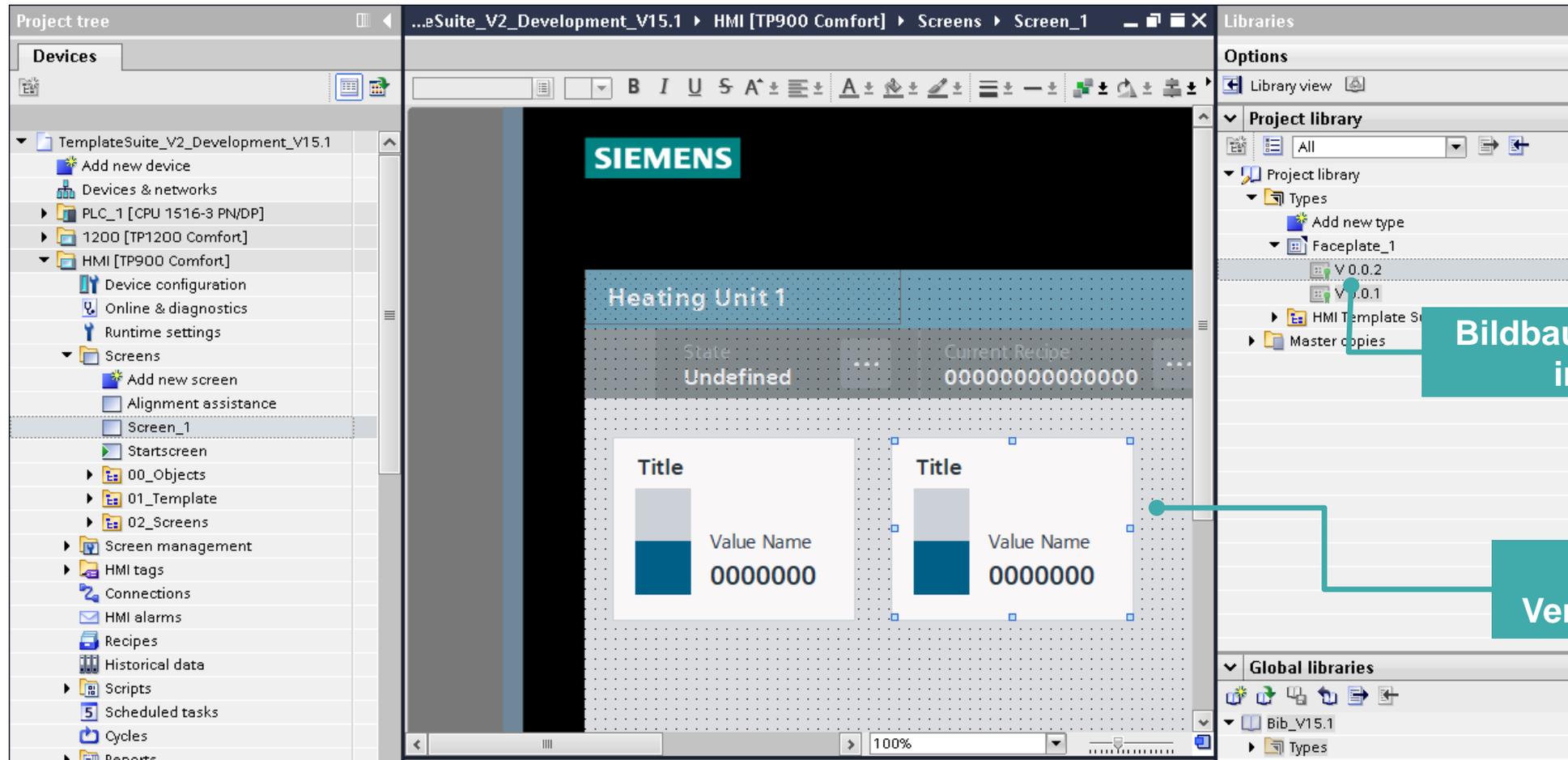
**Automatische Versionsgebung**

**Instanzen im Projekt werden aktualisiert**

**Ungenutzte Versionen werden aus der Bibliothek entfernt**

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines



**Bildbaustein mit Version(en)  
in der Bibliothek**

**Mehrfache  
Verwendung im Bild**

# HMI Design – Bildbausteine

## Erstellen eines Bildbausteines

The screenshot shows the Siemens HMI design environment. At the top, a graphical representation of a tank level indicator is displayed on a grid. The tank is labeled "Tank1" and shows a "Filling Level" of "000000". A teal callout box labeled "Interface auswählen" points to the graphical element.

Below the graphical view is the "Properties" window, which has tabs for "Properties", "Interface", "Animations", "Events", and "Texts". The "Interface" tab is active, showing a table for configuring the interface elements. A teal callout box labeled "Variable verschalten" points to the "Dynamization" column of the table.

Name	Static value	Dynamization
Properties_Faceplate		
tagFilling	0	PLC_Tank_Filling
txtTitle	Tank1	
txtValueName	Filling Level	

A teal callout box labeled "Name vergeben (Statisch)" points to the "Static value" column of the table.

# HMI Design – Bildbausteine

- 1 Was ist ein Bildbaustein?
- 2 Wie werden Bildbausteine erstellt?
- 3 **Welche Vorteile haben Bildbausteine?**

Was sind die  
wesentlichen  
Vorteile?

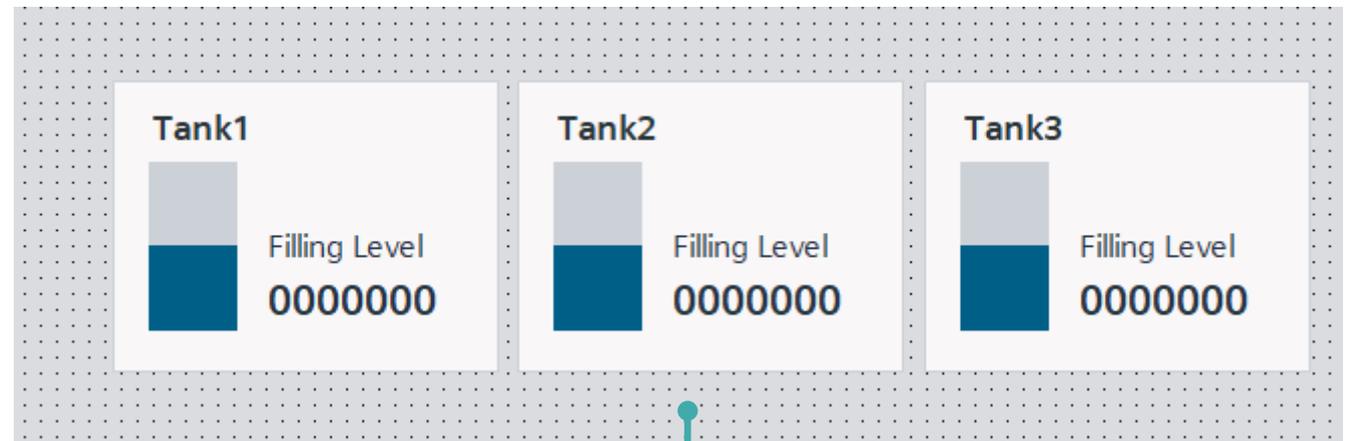


- Änderungen erfolgen an zentraler Stelle.
- „Alte“ & erprobte Varianten können beibehalten werden
- Allgemeine Vorteile durch die Bibliotheksverwaltung (Rückverfolgbarkeit, Austausch, ...)

# HMI Design – Bildbausteine

## Vorteile von Bildbausteinen

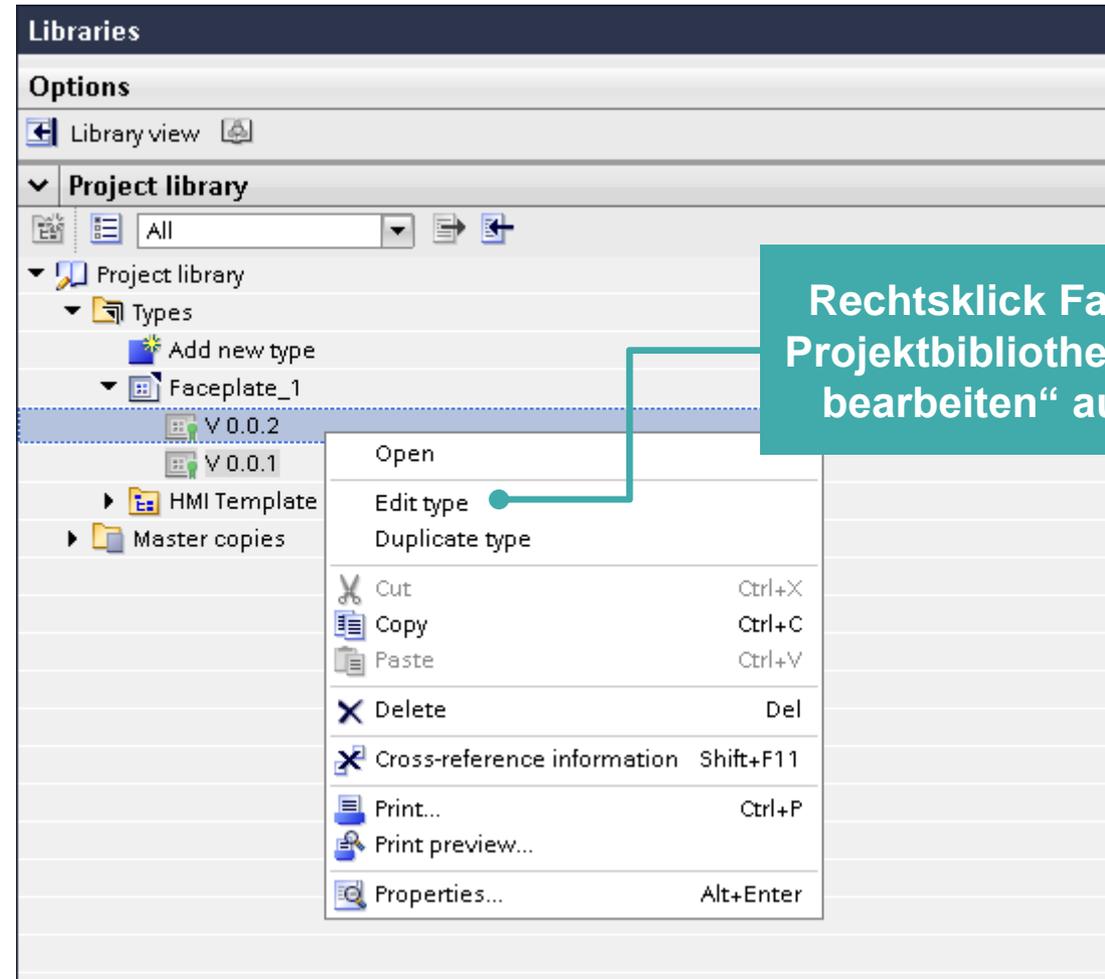
**Beispiel:**  
**Design soll aktualisiert werden**



Durch Verwendung von  
Bildbausteinen ist es nicht  
mehr notwendig jedes Objekt  
einzeln zu bearbeiten

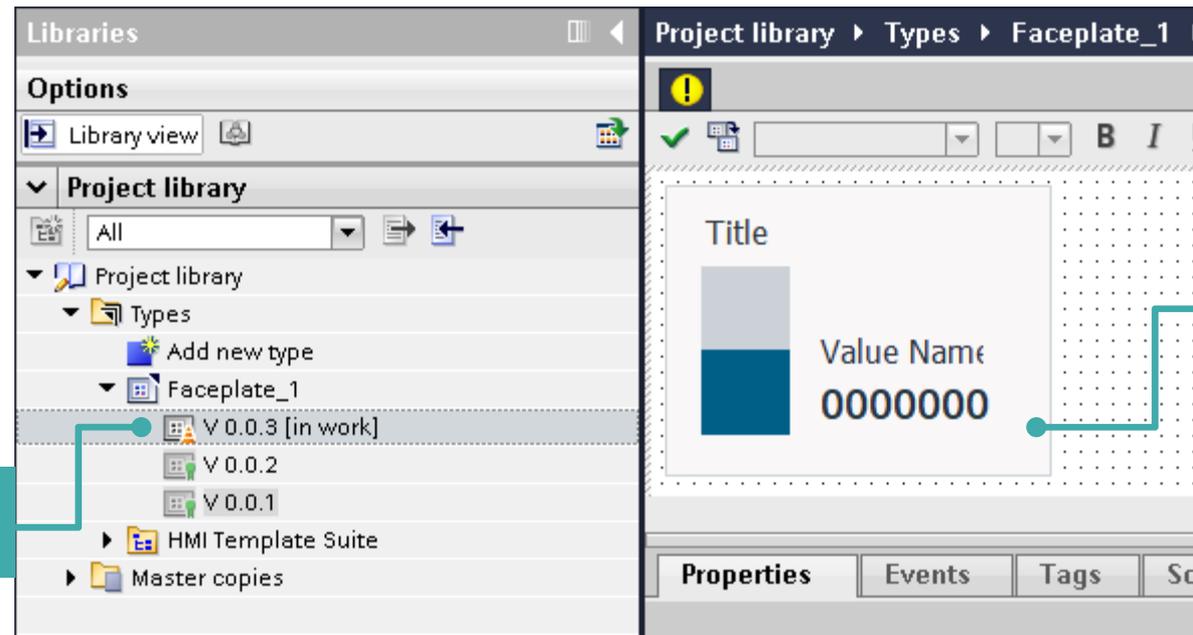
# HMI Design – Bildbausteine

## Vorteile von Bildbausteinen



# HMI Design – Bildbausteine

## Vorteile von Bildbausteinen



Neue Version erstellen  
und bearbeiten

Objekte nach Belieben  
anpassen und/oder neue  
hinzufügen bzw. entfernen.

# HMI Design – Bildbausteine

## Vorteile von Bildbausteinen

Version  
freigeben und  
Instanzen  
aktualisieren

The screenshot displays the 'Libraries' panel in Siemens HMI Design software. The left pane shows a tree view under 'Project library' with a sub-folder 'Types'. Under 'Types', there is a folder 'Faceplate\_1' containing three versions: 'V 0.0.3 [in work]', 'V 0.0.2', and 'V 0.0.1'. A teal box on the left contains the text 'Version freigeben und Instanzen aktualisieren', with a line pointing to the 'V 0.0.3 [in work]' version. The right pane shows the selected version 'V 0.0.3' with a yellow warning banner that reads: 'This type version is currently in the "in work" state. You can [release the version](#) or [discard the changes and](#)'. Below the banner is a preview of a 'Faceplate\_1' widget with a red bar at the bottom and the text 'Value Name 0000000'. At the bottom of the right pane are tabs for 'Properties', 'Events', 'Tags', and 'Scripts'.

# HMI Design – Bildbausteine

## Vorteile von Bildbausteinen

Neue  
Versionsnummer

Aktualisieren der im  
Projekt vorhandenen  
Bildbausteine auf die  
neue Versionsnummer

**Release type version**

**i** Define the properties for the released type version.

A new version will be released for the selected types.  
Assign them common properties or confirm the recommended properties.

Name of type: Faceplate\_1

Version: 0.0 .3

Author: siemens

Comment:

**Options**

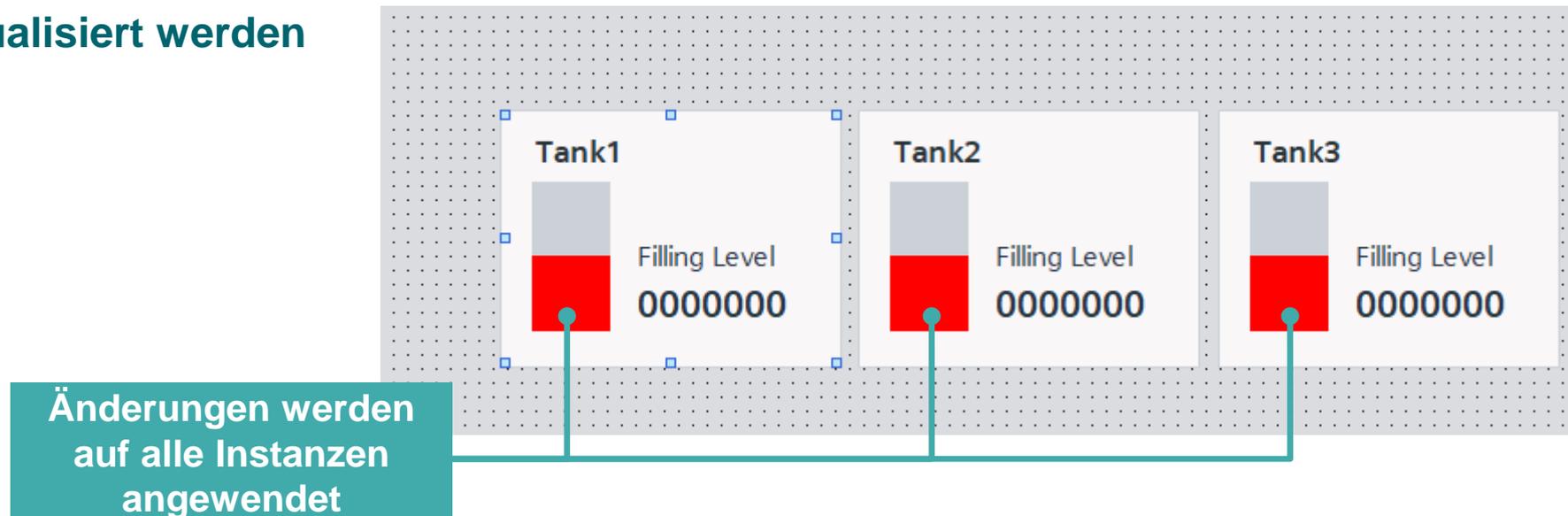
- Update instances in the project
- Delete unused type versions from the library

OK Cancel

# HMI Design – Bildbausteine

## Vorteile von Bildbausteinen

**Beispiel:**  
**Design soll aktualisiert werden**



# HMI Design Stile

## Die Herausforderung – Wiederverwendbarkeit von Objekten

Ist es möglich die Darstellung von  
Objekten zentral zu verwalten?



Ja, mit Hilfe von **Stilen** kann das  
Aussehen von Anzeige- und  
Bedienelemente harmonisiert werden.

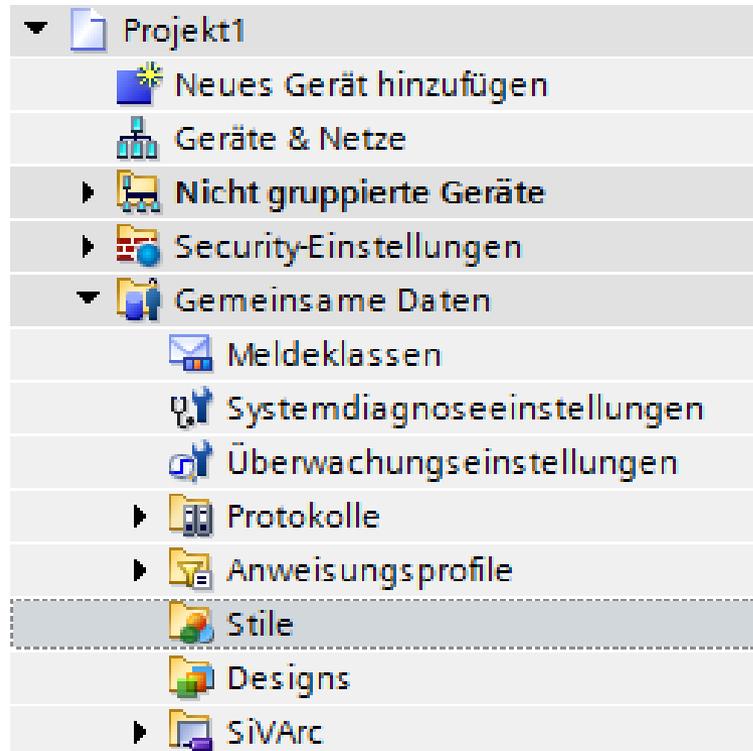


# HMI Design – Stile

- 1 Was ist ein Stil?
- 2 Wie werden Stile erstellt?
- 3 Welche Vorteile haben Stile?

# HMI Design – Stile

## Was ist ein Stil?



## Stile ...

- dienen der Definition des Aussehens von Anzeige- und Bedienelemente.
- werden projektweit angelegt und verwaltet.
- stehen zur Verfügung für WinCC Basic/Comfort/Advanced.
- können versioniert und über die Bibliothek ausgetauscht werden.

# HMI Design – Stile

## Was ist ein Stil?

### Stile im TIA-Portal - Projekt

Stile		
Stand...	Name mit Typ-Version	Kommentar
	HMI Template Suite101 - ...	
	WnCC Dark	
	WnCC Fresh	
	WnCC Light	
	WnCC Wireframe	
	<Hinzufügen>	

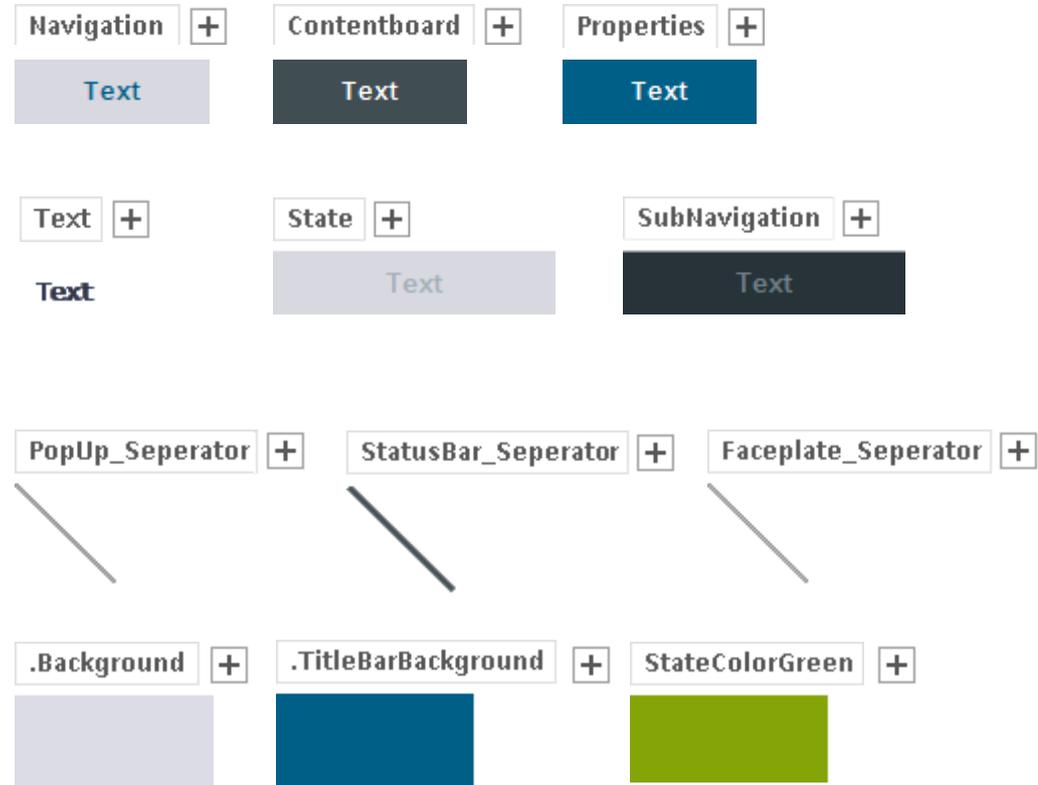
### Beispiel-Objekte im Stil

#### Schaltflächen

#### Texte

#### Polygone/Linien

### Stil-Elemente



# HMI Design – Stile

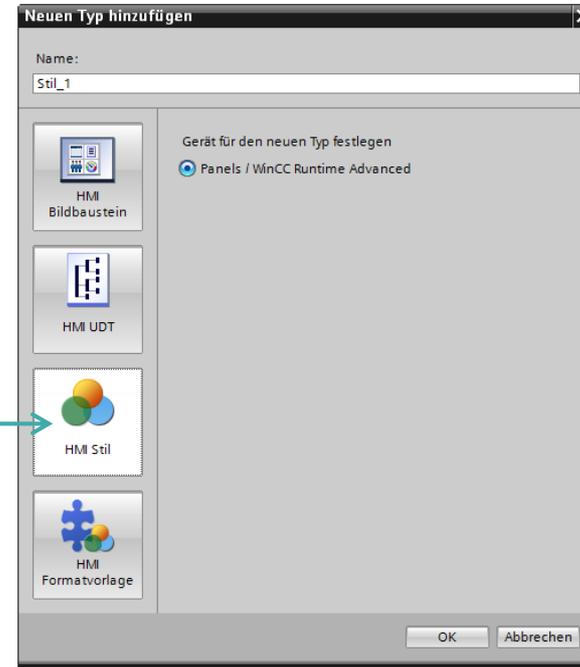
- 1 Was ist ein Stil?
- 2 **Wie werden Stile erstellt?**
- 3 Welche Vorteile haben Stile?

# HMI Design – Stile

## Stile erstellen



### A) Projekt Bibliothek

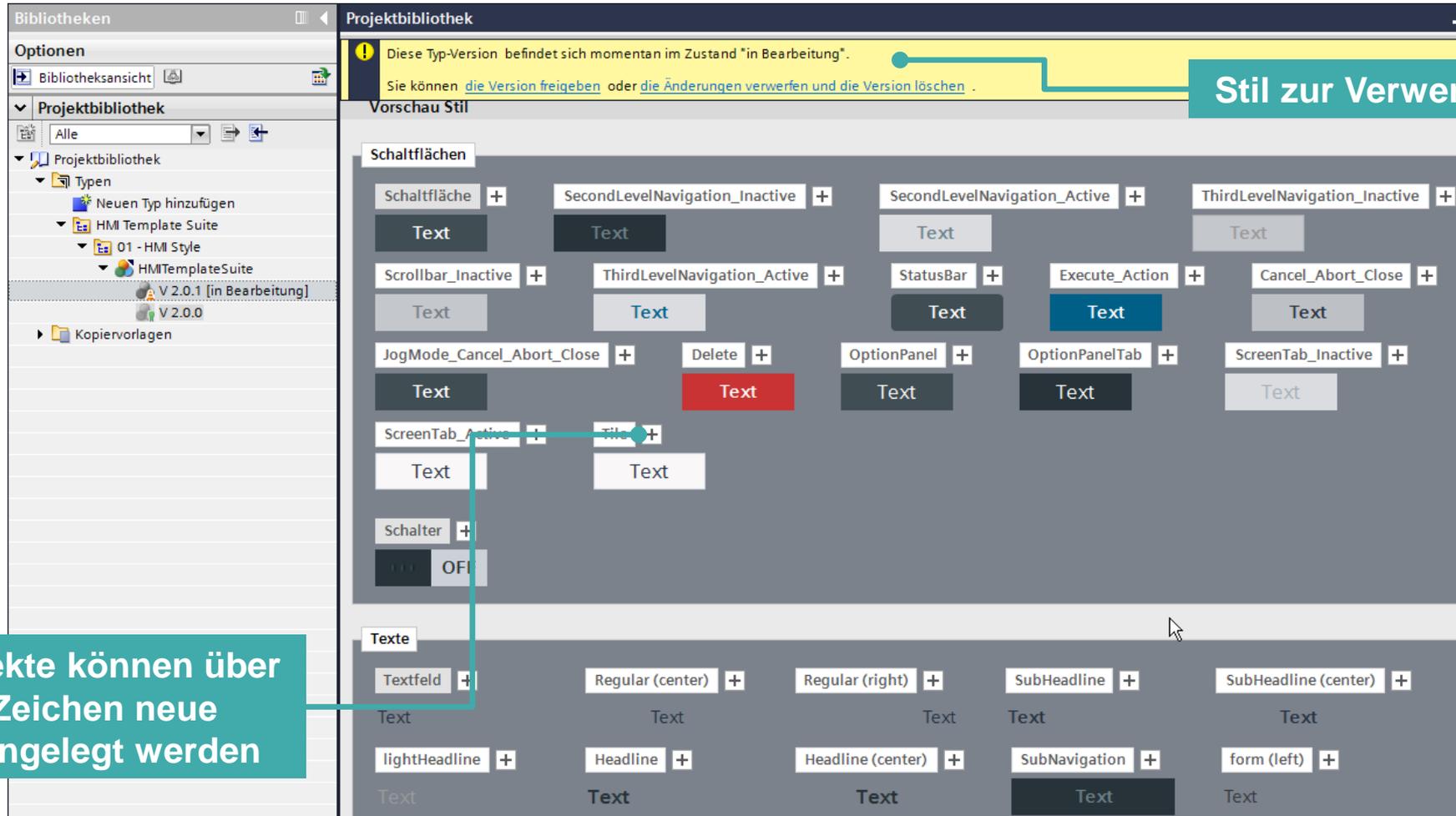


### B) Stile Ordner

Stile			
	Stand...	Name mit Typ-Version	Kommentar
	<input checked="" type="radio"/>	HMI Template Suite101 - ...	
	<input type="radio"/>	WinCC Dark	
	<input type="radio"/>	WinCC Fresh	
	<input type="radio"/>	WinCC Light	
	<input type="radio"/>	WinCC Wireframe	
		<a href="#">Hinzufügen&gt;</a>	

# HMI Design – Stile

## Stile erstellen



Für alle Objekte können über das „+“ Zeichen neue Elemente angelegt werden

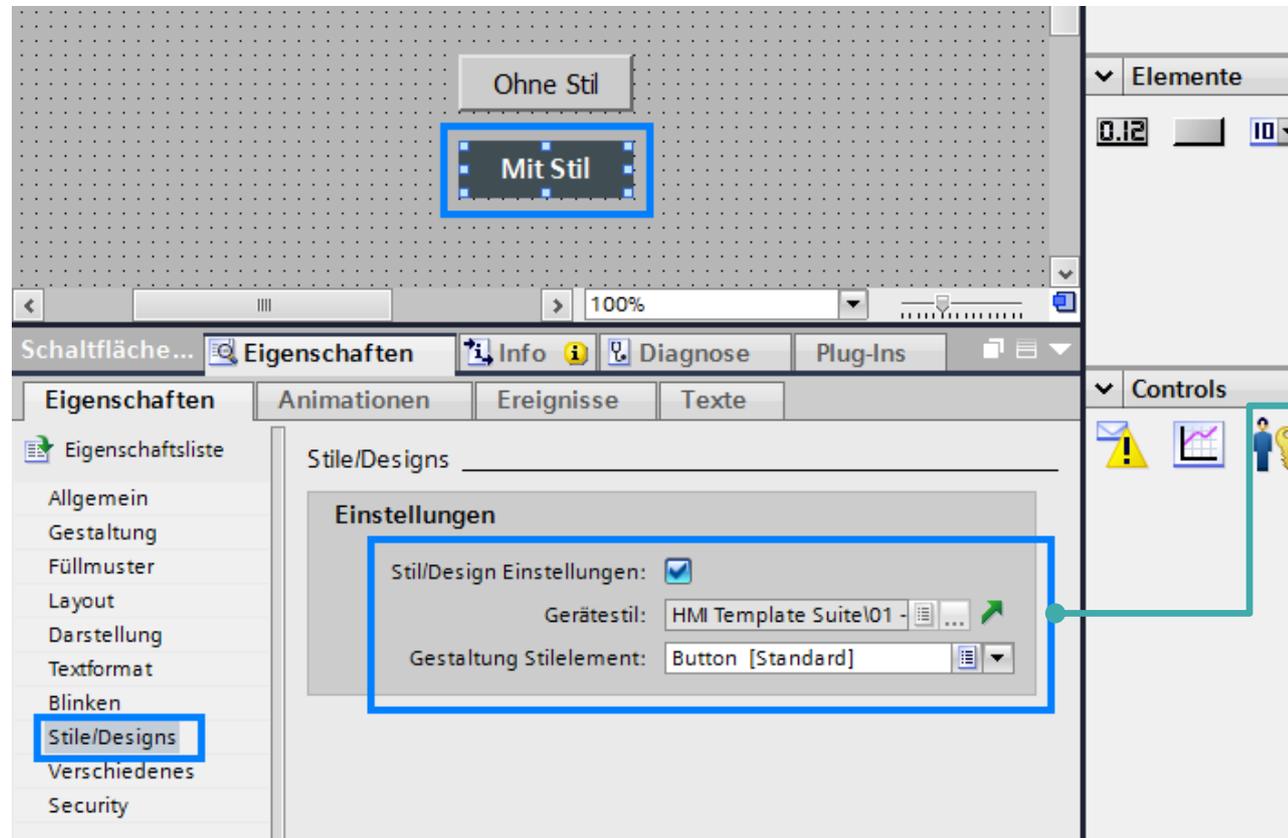
Stil zur Verwendung freigeben

# HMI Design – Stile

- 1 Was ist ein Stil?
- 2 Wie werden Stile erstellt?
- 3 **Welche Vorteile haben Stile?**

# HMI Design – Stile

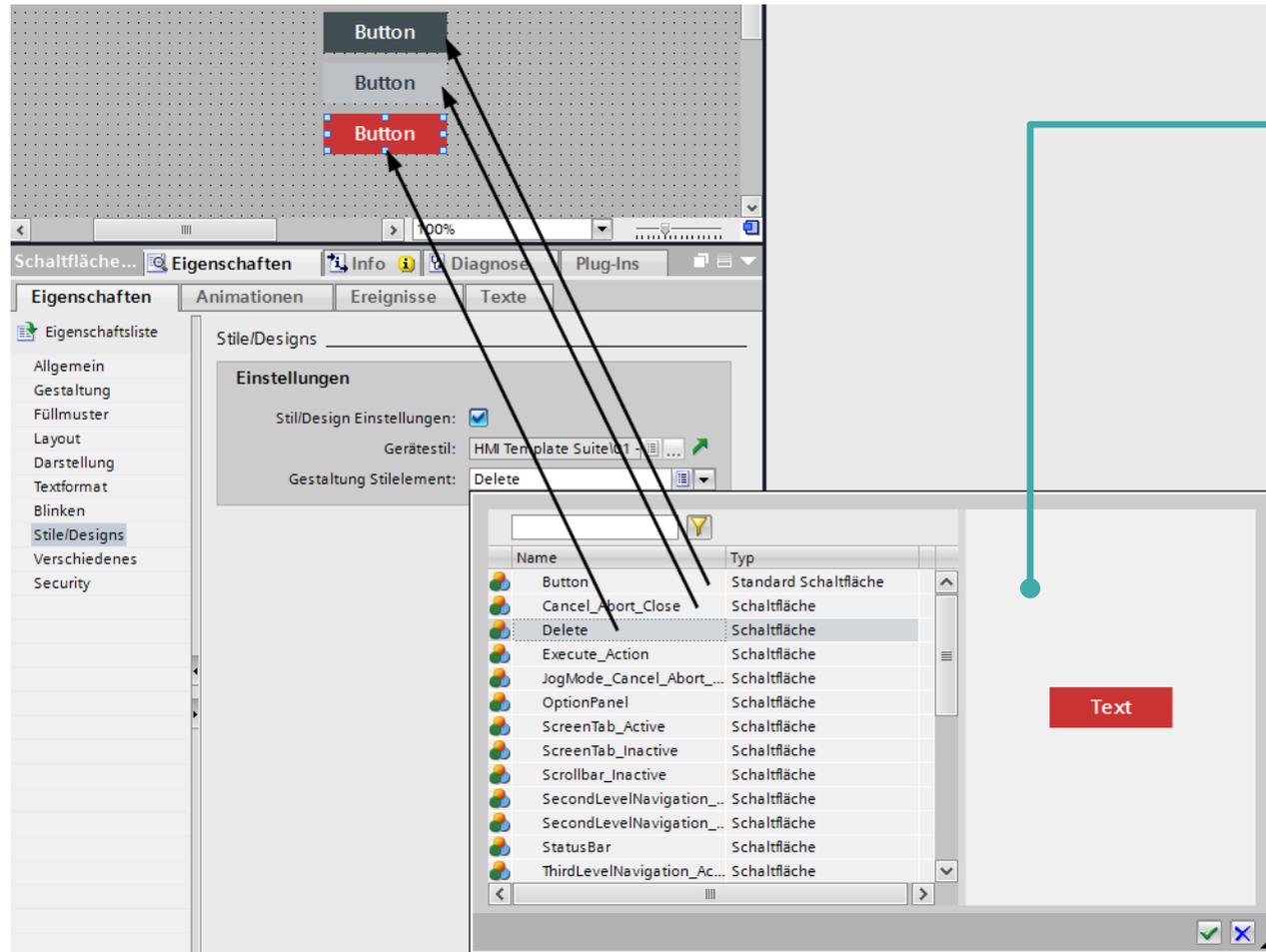
## Vorteile von Stilen



Einheitliches Aussehen  
und Zeiteinsparung durch  
Verwendung von Stilen

# HMI Design – Stile

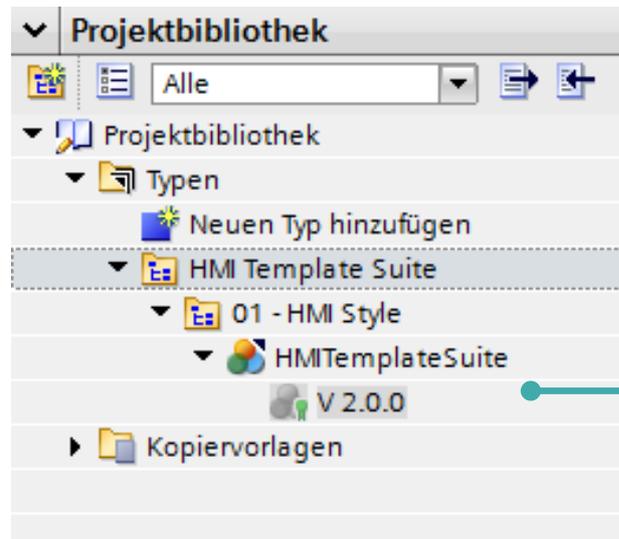
## Vorteile von Stilen



Frei definierbare Stil-Elemente können einfach ausgewählt werden.

# HMI Design – Stile

## Vorteile von Stilen



Stile können versioniert und über die Bibliothek ausgetauscht werden

Was sind die  
wesentlichen  
Vorteile?



- Einheitliches Aussehen
- Zeiteinsparung durch vordefinierte Elemente
- Versionierung und Austausch über die Bibliothek

# HMI Design Grundlagen HMI Design

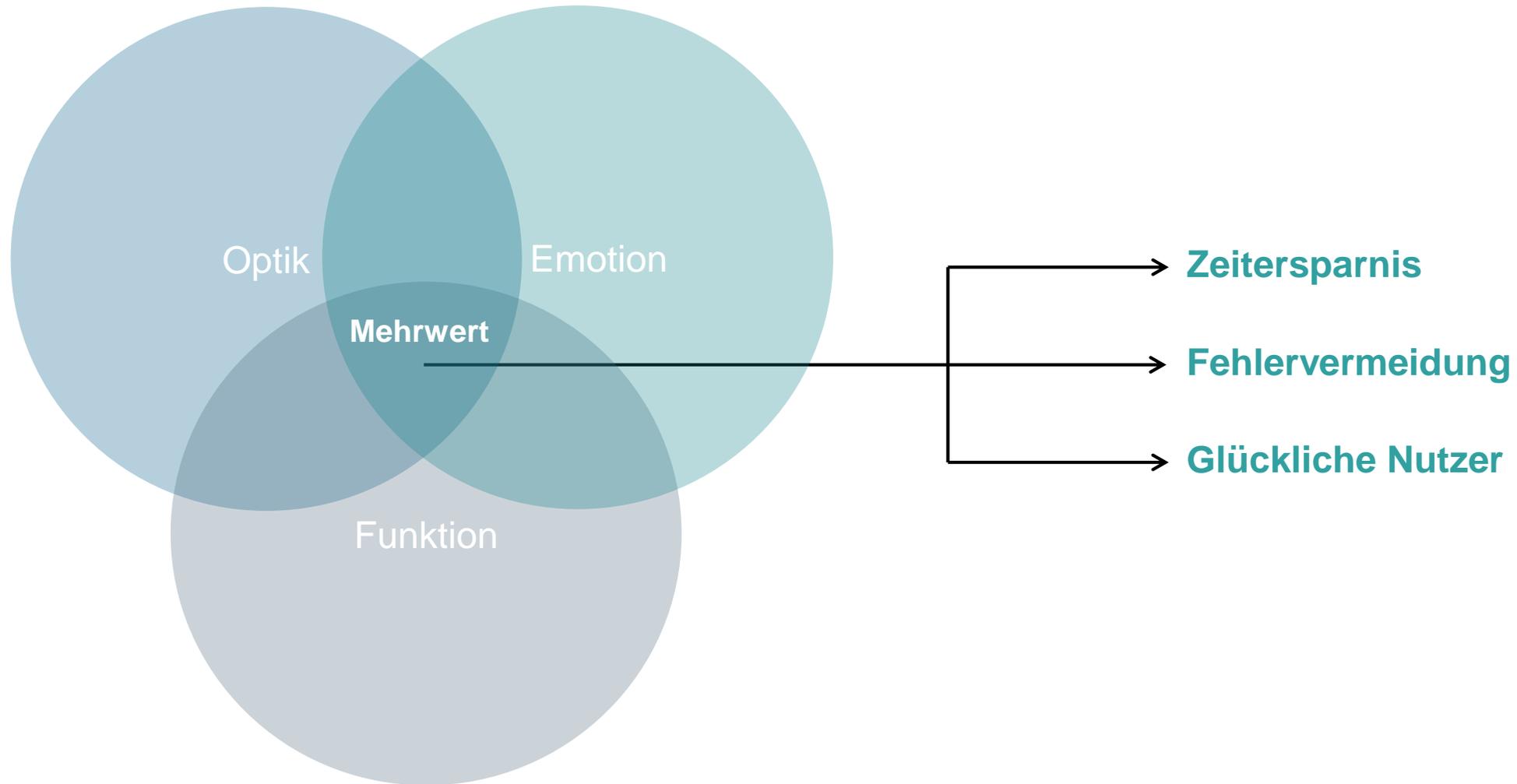
- 1 **Was ist HMI Design?**
- 2 Warum ist HMI Design wichtig?
- 3 Wie wird ein gutes HMI Design entworfen?

# Was ist Design?

**Design is not just what it  
looks like and feels like.  
Design is how it works.**  
steve jobs

cathDSGN.

# Was ist Design? Kernpunkte



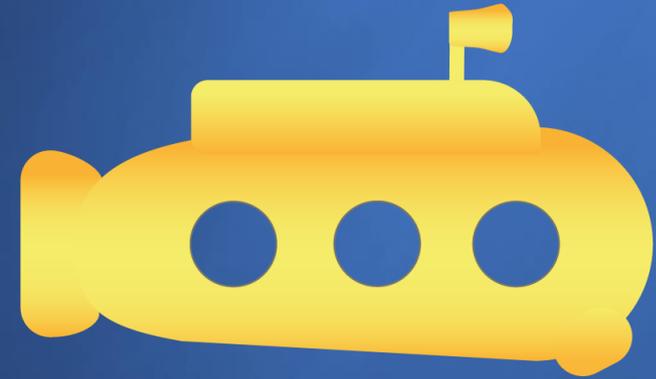
Endkunde

Angezeigter  
Inhalt

HMI

Maschine

Konstruktion  
und  
Funktion



- 1 Was ist HMI Design?
- 2 **Warum ist HMI Design wichtig?**
- 3 Wie wird ein gutes HMI Design entworfen?

**Warum ist HMI Design wichtig?**  
Designstudien belegen, dass...

**SIEMENS**  
*Ingenuity for life*

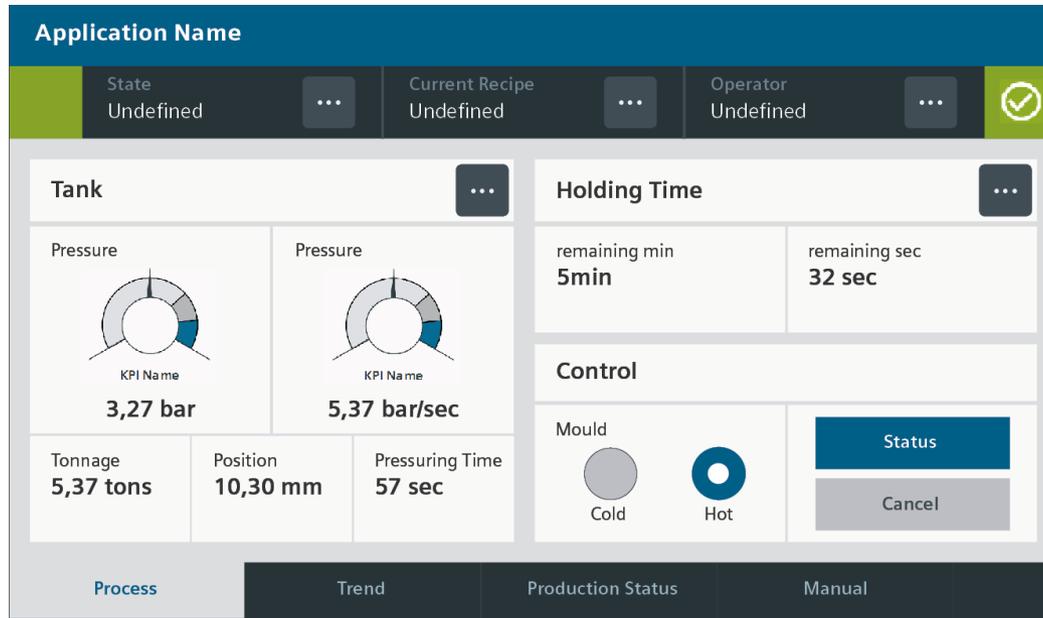
**69,1 % der  
Befragten**

**... geben an, dass Design einen großen Einfluss  
auf die Gesamt-Rendite ihres Unternehmens hat.**

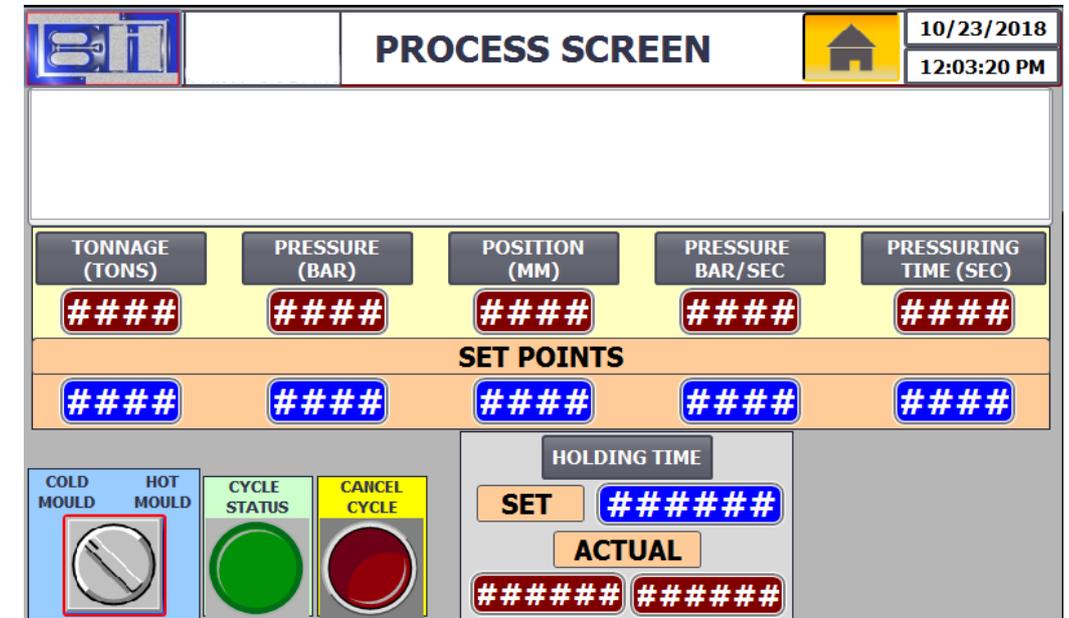
**Rat für Formgebung**  
German Design Council 

# Warum ist HMI Design wichtig? Wirkung

## Maschinenhersteller A



## Maschinenhersteller B



Mit welchem Screen verbindet der Endkunde Professionalität?

**Im Wettbewerb ist das Thema Design nicht zu unterschätzen und nimmt zunehmend eine wichtige Rolle ein.**



- 1 Was ist HMI Design?
- 2 Warum ist HMI Design wichtig?
- 3 **Wie wird ein gutes HMI Design entworfen (Design Workflow)?**

# Design Workflow Überblick



## GESTALTUNG

Farben, Icons, Fonts, etc.

## AUFBAU

Screen-Typen, Navigation, etc.

## STRUKTUR

Information Architecture, User Flow

## FOKUS

Einsatzbereiche, Zielgruppe

## STRATEGIE

Use Cases, Kundenanforderungen

## 1. STRATEGIE

- Sammlung der Kundenanforderungen
- Interviews / Research  
„Was braucht der Bediener?“
- Definition der Use Cases zur Bedienung



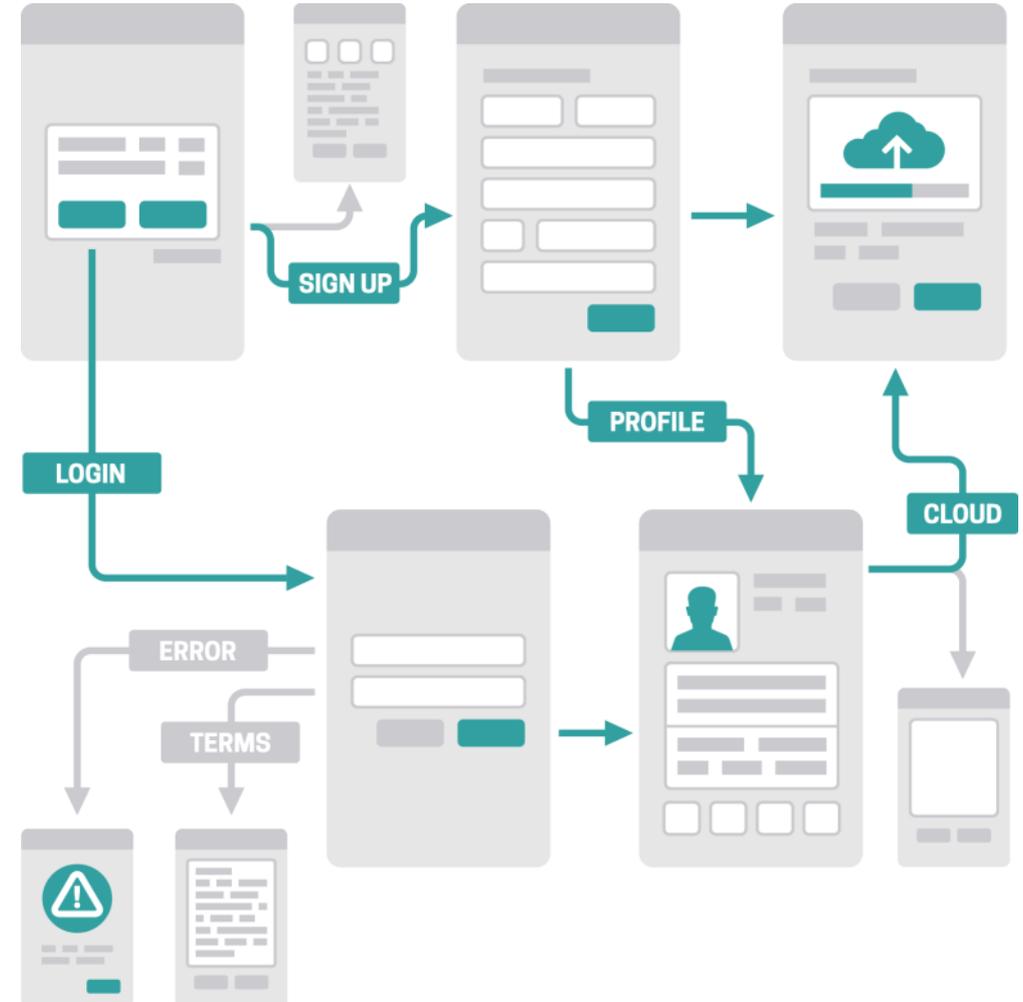
## 2. FOKUS

- Einsatzbereiche definieren  
Near- / Farfield, In- / Outdoor, etc.
- Zielgruppe definieren  
Welches Personal bedient die Maschine  
(Profil erstellen)?
- Handschuhbedienung?
- Use Cases priorisieren



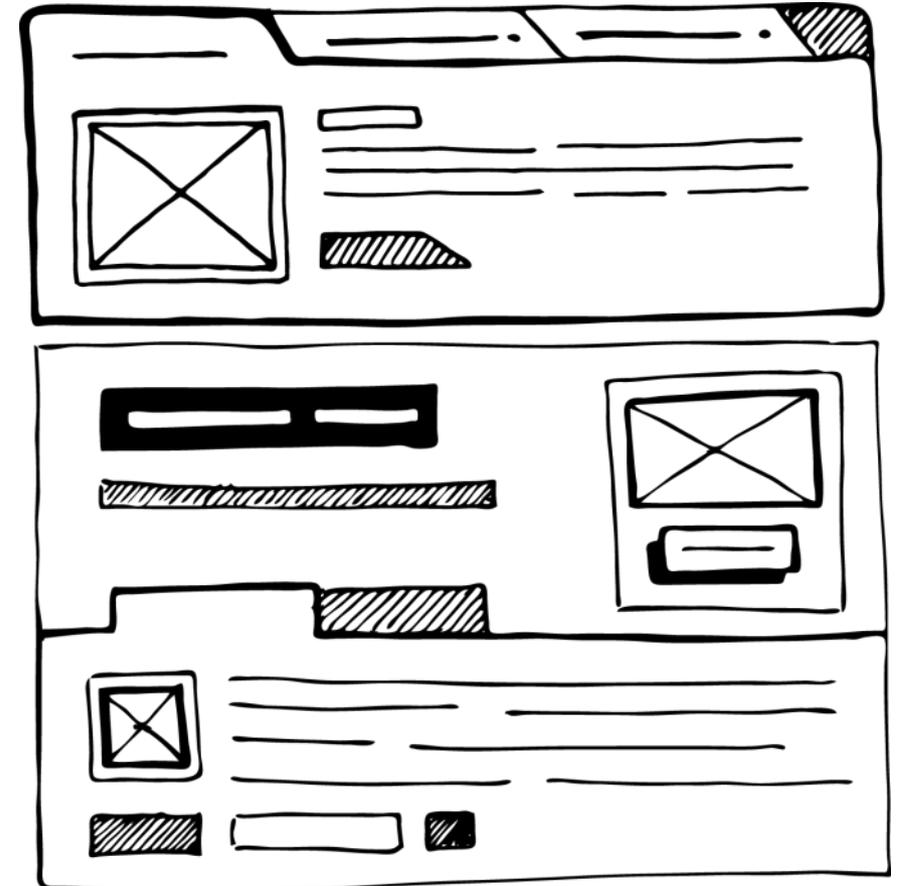
## 3. STRUKTUR

- Information Architecture  
Wo werden welche Informationen benötigt?
- Hierarchie der einzelnen Screens festlegen
- „User Flow“ gestalten



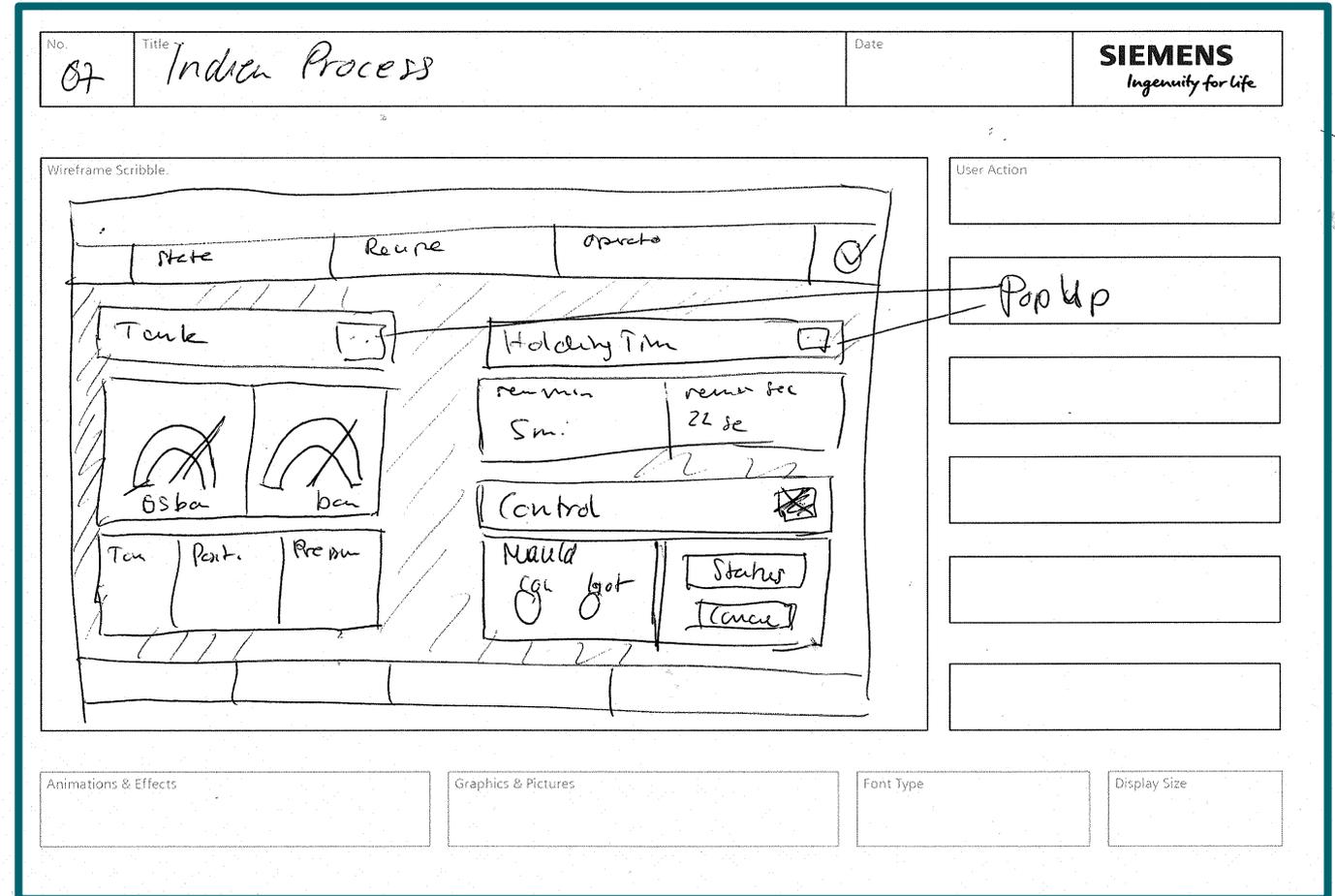
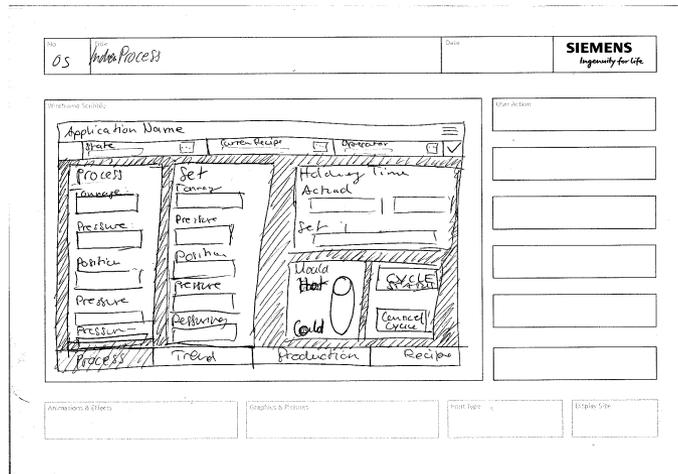
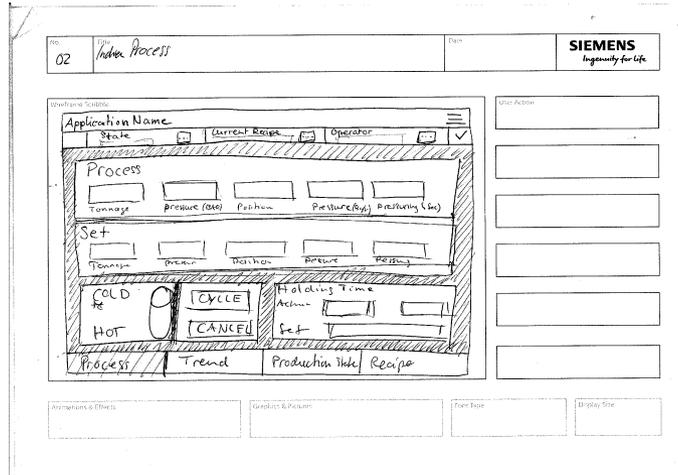
## 4. AUFBAU

- Definition der Screen-Typen (Design Patterns)
- Aufbau und Hierarchie der Screens
- Welche Objekte werden für welche Aktion verwendet (Interaktion).



# Design Workflow

## Schnelles Feedback durch Scribbles



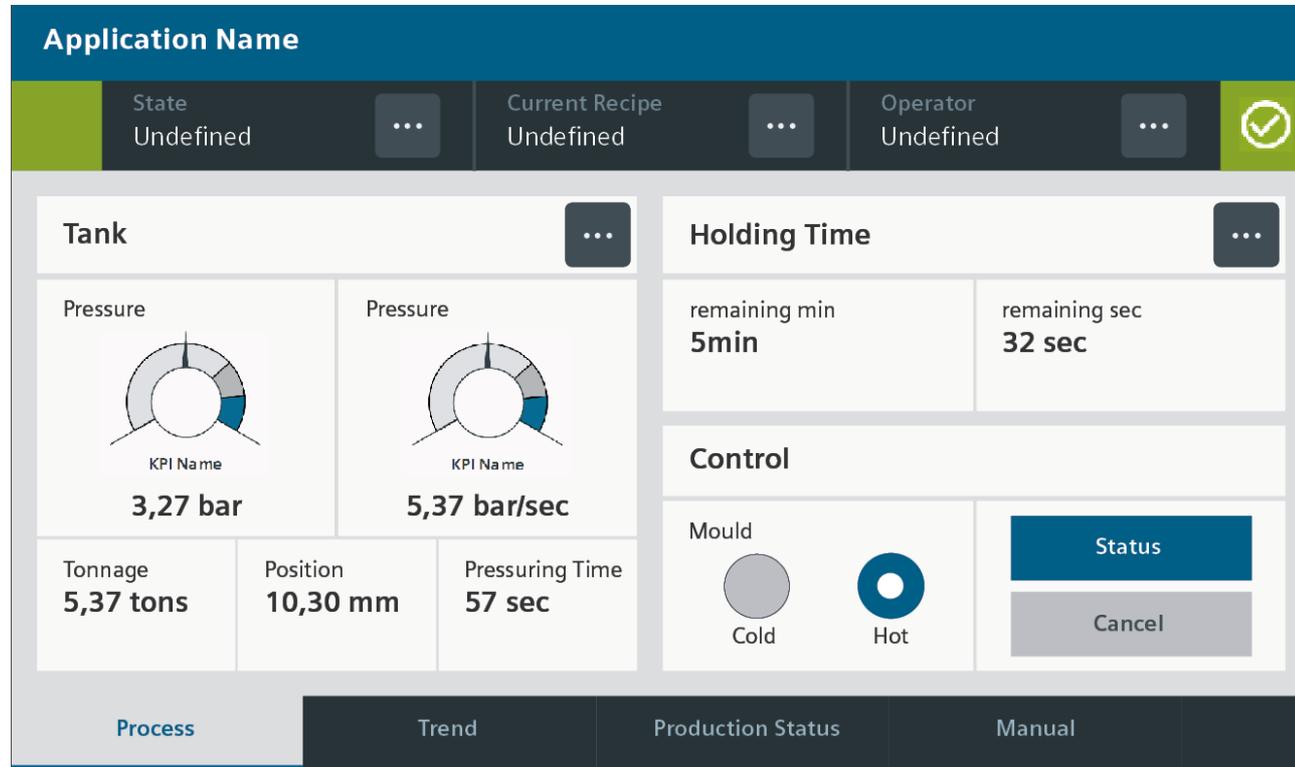
## 5. GESTALTUNG

- Definition der Farben  
(Corporate Design Vorgaben)
- Definition der Icons
- Gestaltung der Objekte
- Grafische Ausarbeitung der Screens



# Design Workflow

## Unsere Designlösung



### Benutzerfreundliches HMI

- Ganzer Screen sinnvoll genutzt
- Hierarchie durch Überschriften und Content Boxen
- Klickbare Elemente sind klar erkennbar
- Eingabe- und Ausgabefelder durch Pop-Ups getrennt
- Farben unterstützen das Zurechtfinden und lenken nicht unnötig ab
- Konsistente Screens

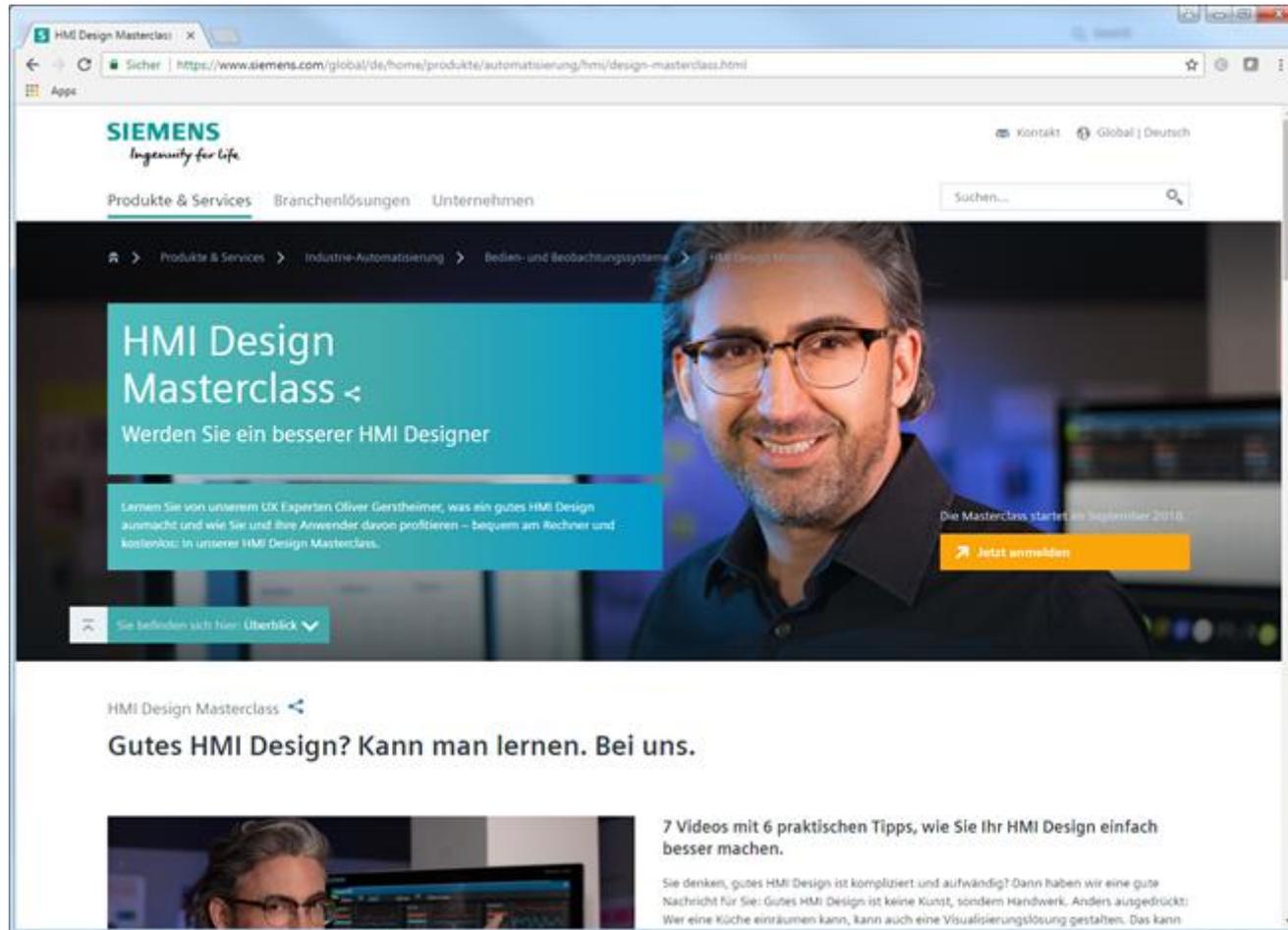
# HMI Design Masterclass

Werden Sie ein besserer HMI Designer



# SIMATIC HMI Design Masterclass

**SIEMENS**  
*Ingenuity for life*



HMI Design Masterclass

Werden Sie ein besserer HMI Designer

Lernen Sie von unserem UX Experten Oliver Gerstheimer, was ein gutes HMI Design ausmacht und wie Sie und Ihre Anwender davon profitieren – bequem am Rechner und kostenlos: In unserer HMI Design Masterclass.

Die Masterclass startet im September 2018.

[Jetzt anmelden](#)

Sie befinden sich hier: [Überblick](#)

HMI Design Masterclass

Gutes HMI Design? Kann man lernen. Bei uns.

7 Videos mit 6 praktischen Tipps, wie Sie Ihr HMI Design einfach besser machen.

Sie denken, gutes HMI Design ist kompliziert und aufwändig? Dann haben wir eine gute Nachricht für Sie: Gutes HMI Design ist keine Kunst, sondern Handwerk. Anders ausgedrückt: Wer eine Küche einräumen kann, kann auch eine Visualisierungslösung gestalten. Das kann

**In 7x10 Minuten zum besseren HMI Designer werden**

Kostenlose Registrierung zum Video-Kurs:  
[www.siemens.de/hmi-design-masterclass](https://www.siemens.de/hmi-design-masterclass)

## HMI Design ist keine Kunst, sondern ein Handwerk

Kostenloser Download oder Bestellung  
(solange Vorrat reicht):

<https://new.siemens.com/global/de/produkt/e/automatisierung/simatic-hmi/hmi-design-workbook-ebook-download.html>



# HMI Template Suite

Das Designsystem für eine perfekte  
Benutzerfreundlichkeit

Frei verwendbar

[siemens.com/hmi-template-suite](https://www.siemens.com/hmi-template-suite)

Ideengeber Mehrwert

**Design-Vorlage**

Flexibel änderbar

**Einheitlichkeit** Reduzierung Aufwand

Integration bestehender Applikationen

**Verfügbarkeit in allen Größen**

**Zeitersparnis**

# HMI Template Suite – Key Facts

Modernes Design-System für industrielle  
Umgebungen optimiert

Entwickelt mit einer  
professionellen Designagentur

Jedes Element  
ist vollständig anpassbar

Verfügbar für verschiedene  
Bildschirmauflösungen

TIA Portal Library  
für einfache Handhabung

**Kostenloses  
Anwendungsbeispiel**

# HMI Template Suite – Grundstruktur



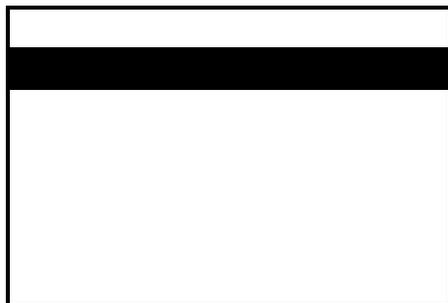
Title Bar



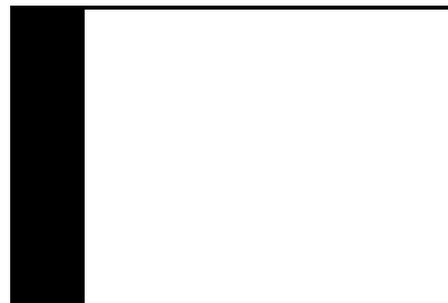
Erste Navigation



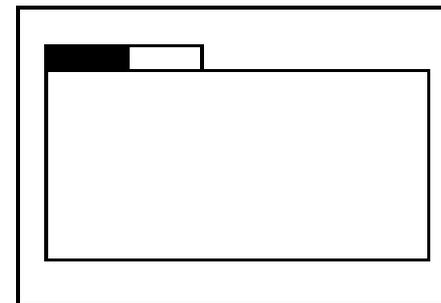
Sub- Navigation



Status Bar



Dritte Navigation



Vierte Navigation

Title Bar für Applikationsnamen  
und öffnen der Hauptnavigation

Maschinenstatus, Benutzer- und  
Rezeptur-Verwaltung über  
Statusbar

4 Verschiedene Navigations-  
Ebenen

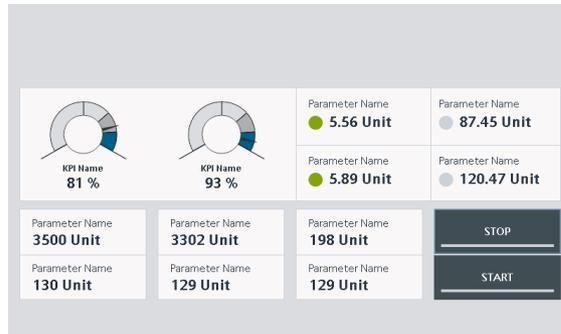


Navigationseben sind optional  
verwendbar

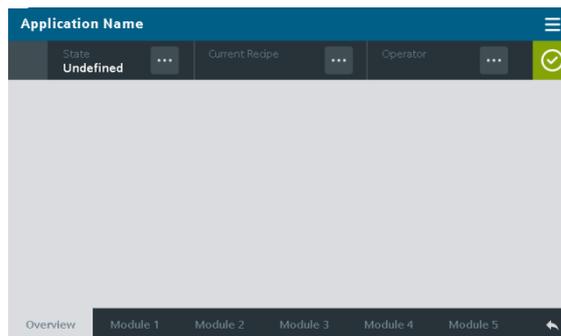
Navigationsebenen sind in Templates vorhanden

# HMI Template Suite – Navigation zuweisen

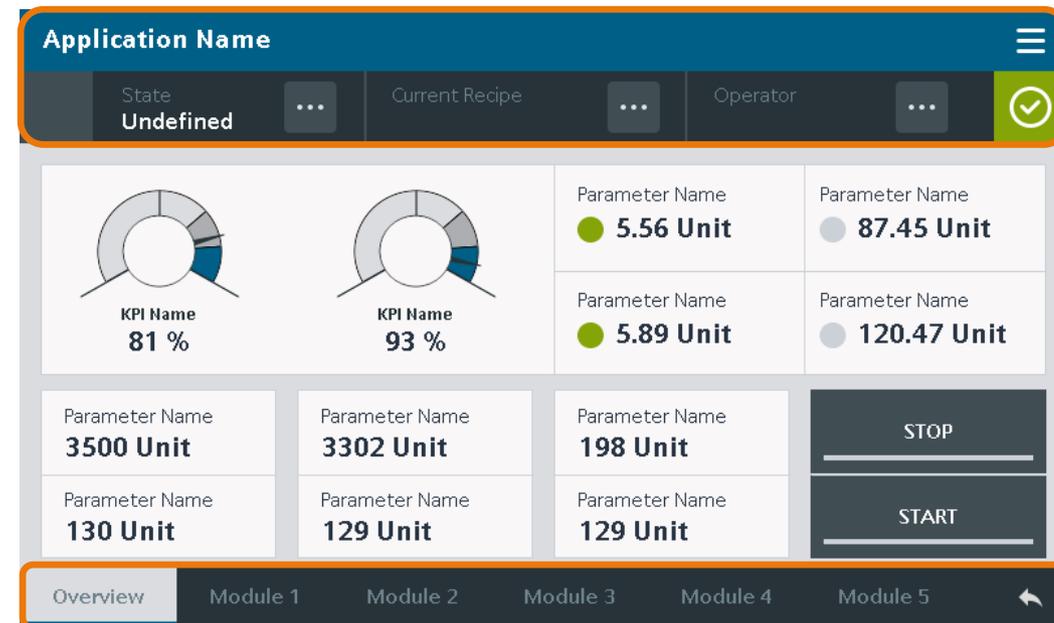
 Bild („Screen“) mit Inhalt



 Vorlage („Screen Template“) mit Navigation



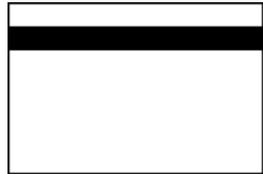
Navigationsebenen erzeugen durch Zuweisung  
der Vorlage zum Bild



# HMI Template Suite – Templates



Title Bar

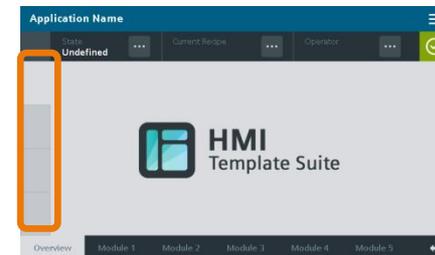
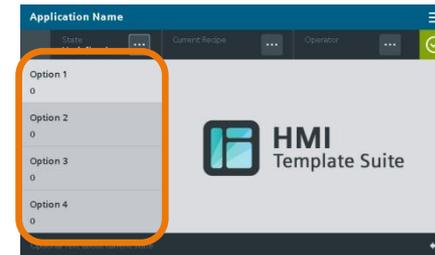
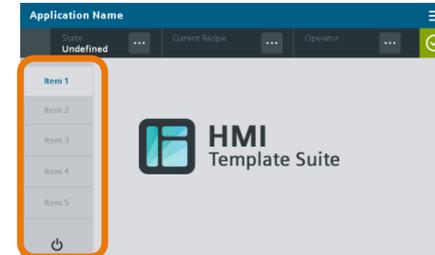
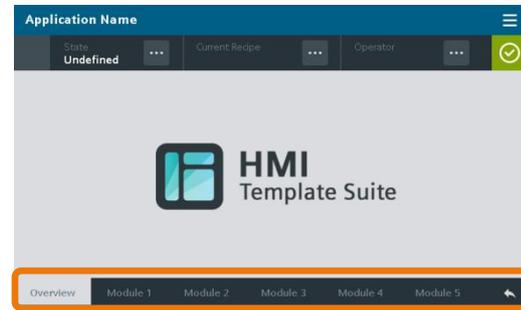


Status Bar

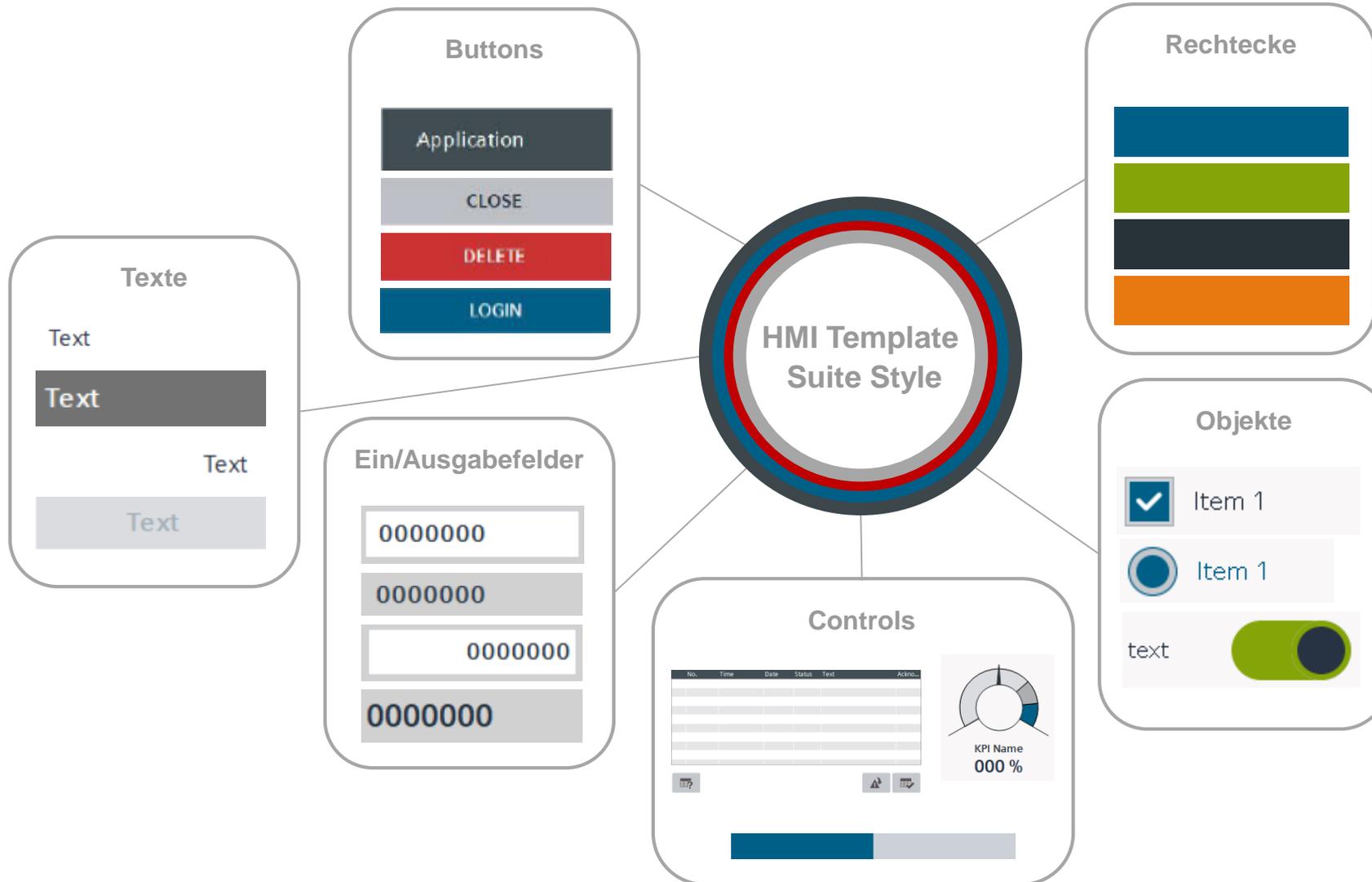


Template mit Subnavigation

In allen Templates enthalten



# HMI Template Suite – HMI Style zur einheitlichen Gestaltung

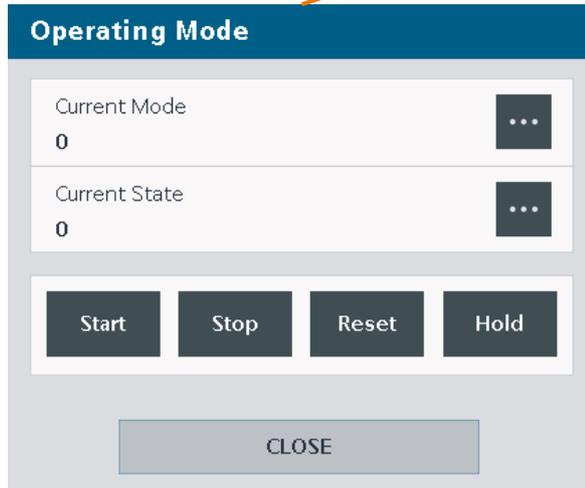
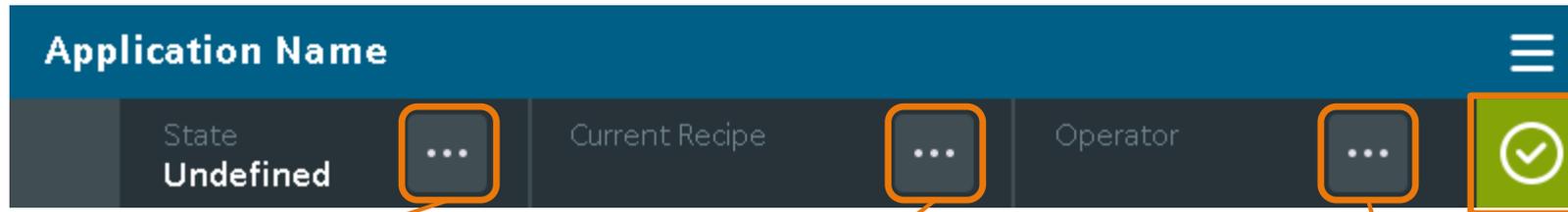


HMI Style zur einheitlichen Gestaltung

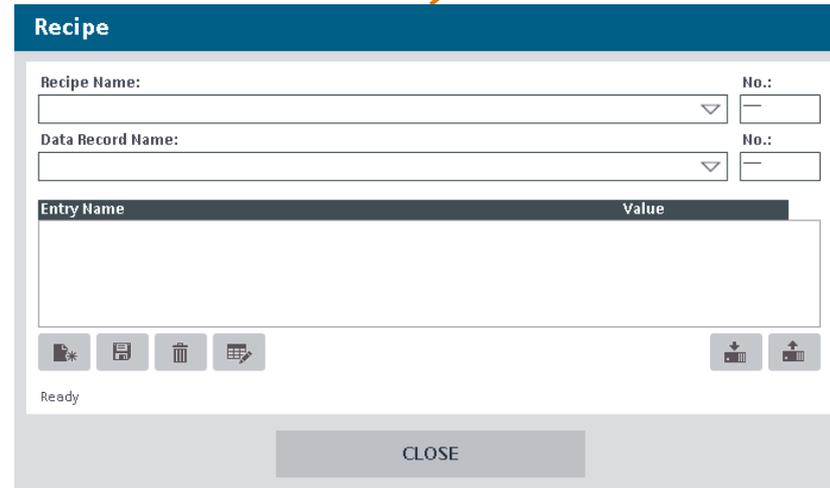
Stil beinhaltet für Objekte viele verschiedene Unter-Stile  
**!** (ein paar Beispiele in der Folie)

Verwendungszweck der Unter-Stile sind in Dokumentation beschrieben

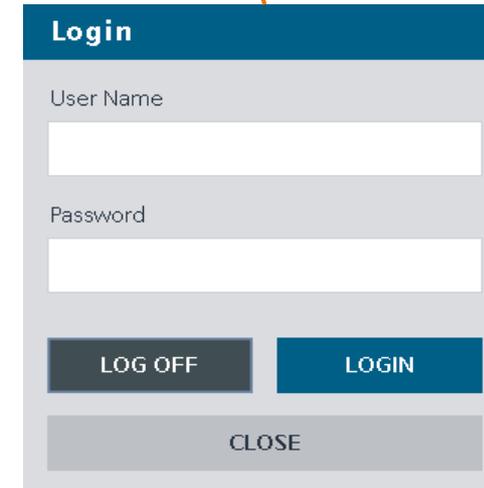
# HMI Template Suite – Basis Funktionen



PopUp Maschinen Status steuern



PopUp für Rezepturen



PopUp Benutzer Login/Logoff



# HMI Template Suite – Beispiel Screens als Vorlage

Application Name

State: Undefined ... Current Recipe: ... Operator: ...

Nr.	Value	
0	0	...
0	0	...
0	0	...
0	0	...
0	0	...
0	0	...
0	0	...
0	0	...

Overview | Module 1 | Module 2 | Module 3 | Module 4 | Module 5

Application Name

State: Undefined ... Current Recipe: ... Operator: ...

Tab 1 | Tab 2

Parameter 1 0	Parameter 2 0	Parameter 3 0	Parameter 4 0
Parameter 5 0	Parameter 6 0	Parameter 7 0	Parameter 8 0

Overview | Module 1 | Module 2 | Module 3 | Module 4 | Module 5

Application Name

State: Undefined ... Current Recipe: ... Operator: ...

KPI Name  
81 %

KPI Name  
93 %

Parameter Name  
5.56 Unit

Parameter Name  
87.45 Unit

Parameter Name  
3500 Unit

Parameter Name  
3302 Unit

Parameter Name  
198 Unit

STOP

Parameter Name  
130 Unit

Parameter Name  
129 Unit

Parameter Name  
129 Unit

START

Overview | Module 1 | Module 2 | Module 3 | Module 4 | Module 5

Application Name

State: Undefined ... Current Recipe: ... Operator: ...

Image of machine module

Parameter 1  
0.0 Unit

Parameter 2  
0.0 Unit

Parameter 3  
0.0 Unit

Parameter 4  
0.0 Unit

Overview | Module 1 | Module 2 | Module 3 | Module 4 | Module 5

Application Name

State: Undefined ... Current Recipe: ... Operator: ...

Step 1 | Step 2 | Step 3 | Step 4 | Step 5

CANCEL

< >

Application Name

State: Undefined ... Current Recipe: ... Operator: ...

KPI Name  
0.00 %

KPI Name  
0.00 %

KPI Name  
0.00 %

KPI Name  
0.00 %

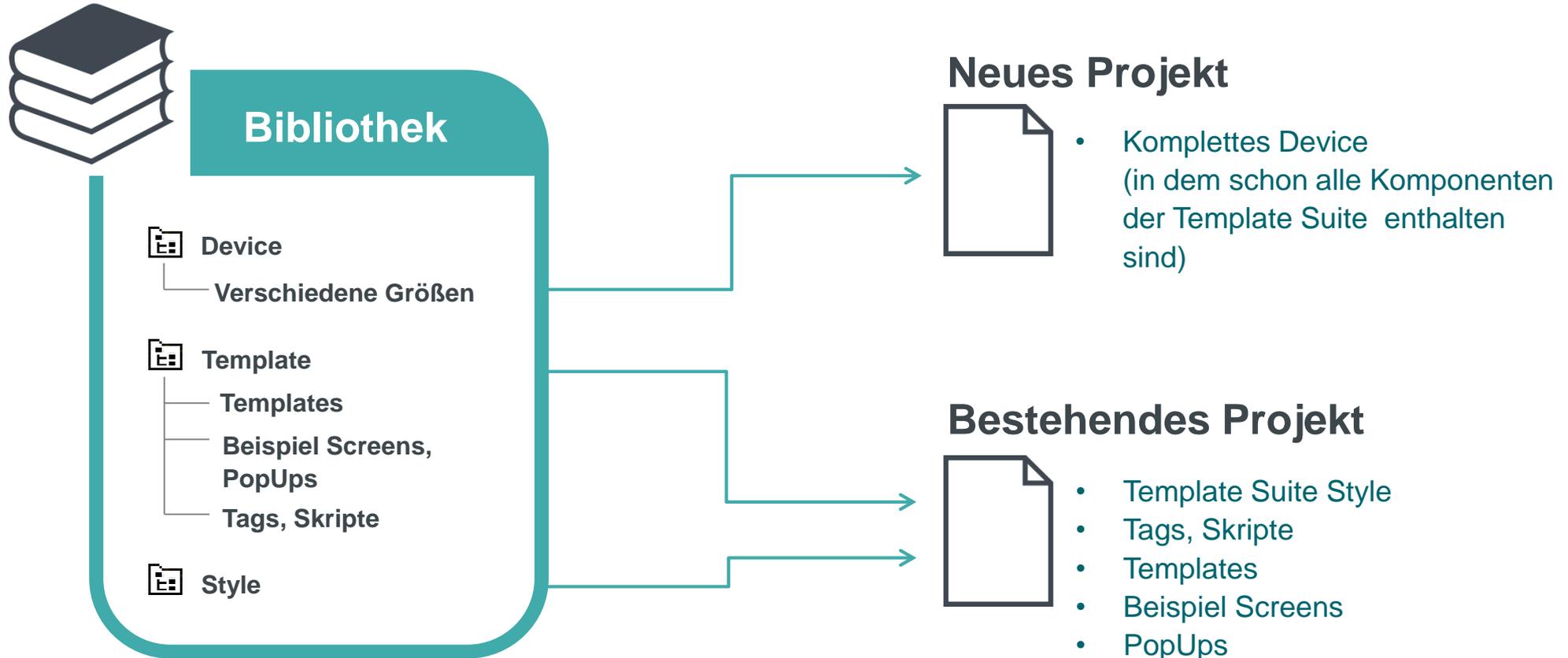
Title  
Subtitle

Title  
Subtitle

Title  
Subtitle

Overview | Module 1 | Module 2 | Module 3 | Module 4 | Module 5

# HMI Template Suite – Integration Template Suite in TIA Portal



Project tree

Devices

- Templae Suite Demo
  - Add new device
  - Devices & networks
  - Ungrouped devices
  - Security settings
  - Common data
  - Documentation settings
  - Languages & resources
  - Online access
  - Card Reader/USB memory

Templae Suite Demo > Devices & networks

Topology view Network view Device view

Network Connections HMI connection Relations

100%

Properties Info Diagnostics

General

No 'properties' available.  
No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

Hardware catalog

Options

Catalog

- Controllers
- HMI
- PC systems
- Drives & starters
- Network components
- Detecting & Monitoring
- Distributed I/O
- Power supply & distribution
- Field devices
- Other field devices

Information

Device:

Article no.:

Version:



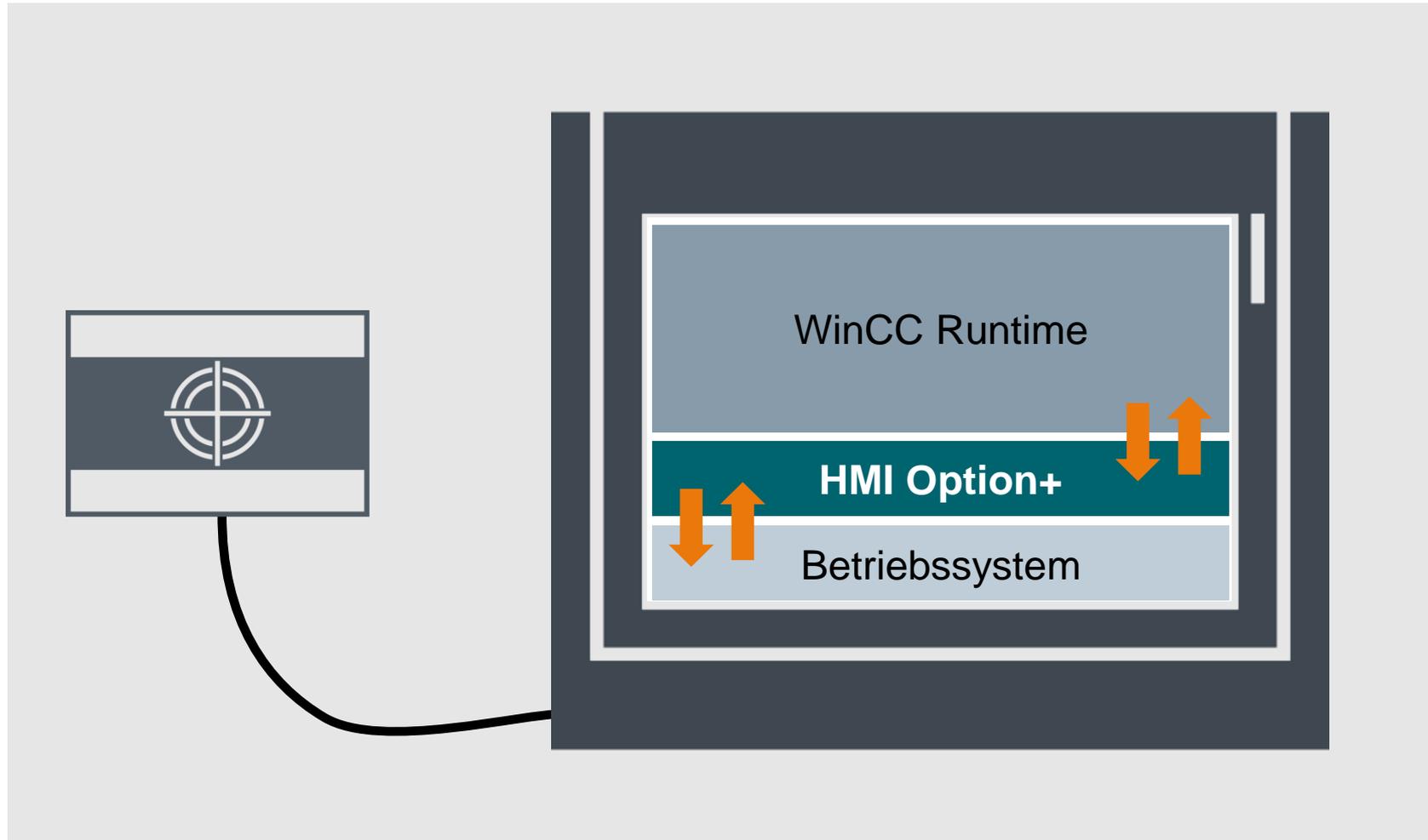
# HMI Option+

Das Plus an Optionen für  
SIMATIC HMI Comfort Panels

# HMI Option+

## Technisches Konzept

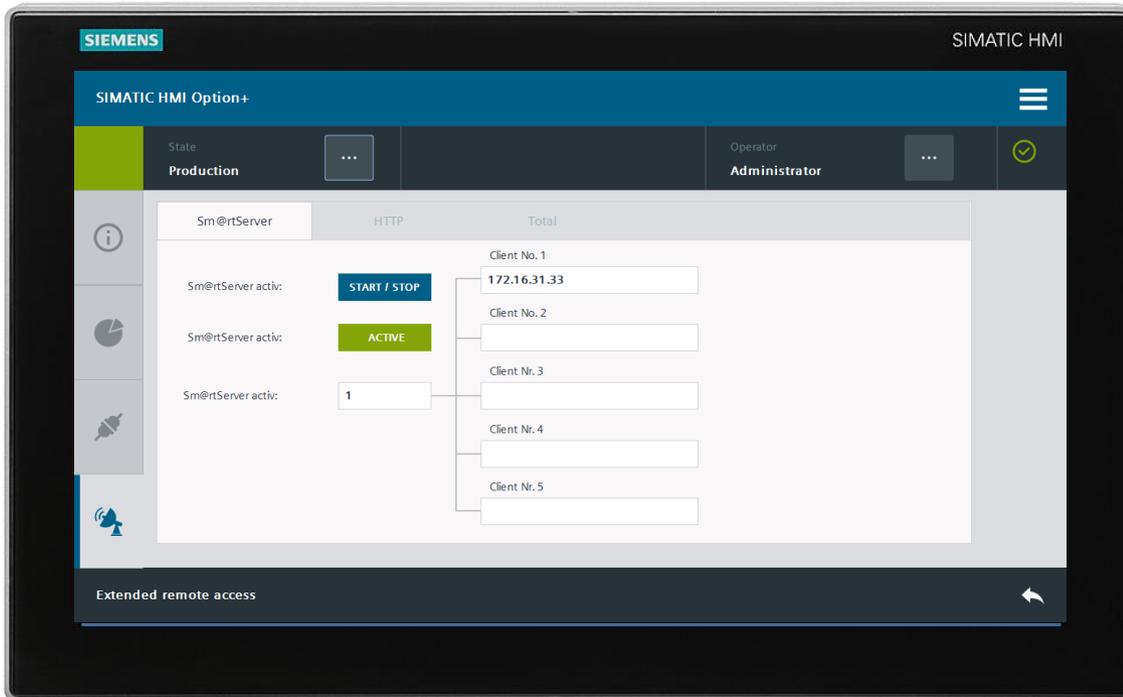
**SIEMENS**  
*Ingenuity for life*



HMI Option+ bildet die Schnittstelle zwischen dem Betriebssystem und der WinCC Runtime zum Austausch von Daten in beide Richtungen. Die Daten stehen in der WinCC Runtime als interne Variablen zur Verfügung.

# HMI Option+ Bestandteile – Runtime Visualisierung

**SIEMENS**  
*Ingenuity for life*



## HMI Option+ Runtime Visualisierung



Moderne, für Touch-Bedienung optimierte  
Visualisierung der Funktionen



Erlaubt Zugriff auf nahezu alle Funktionen von  
Option+ aus der WinCC Runtime heraus



Anpassung an die Anforderungen des Kunden  
möglich

# HMI Option+ Download

**SIEMENS**  
*Ingenuity for Life*



Das vollständige Applikationsbeispiel beinhaltet:

- Installation von Option+
- Dokumentation
- TIA Portal Bibliothek
- TIA Portal Beispielprojekt

Download unter der Beitrags ID **109754400** oder dem folgenden Link aus dem Industry Online Support heruntergeladen werden:

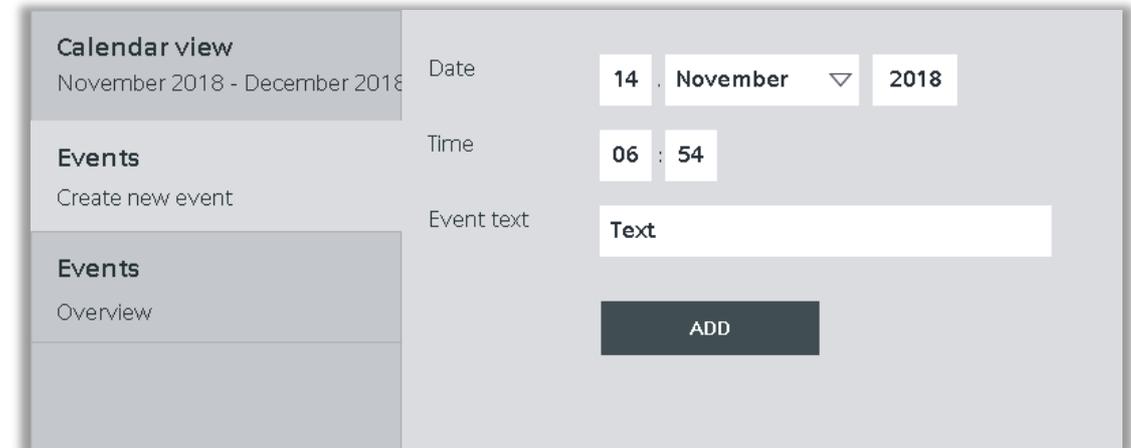
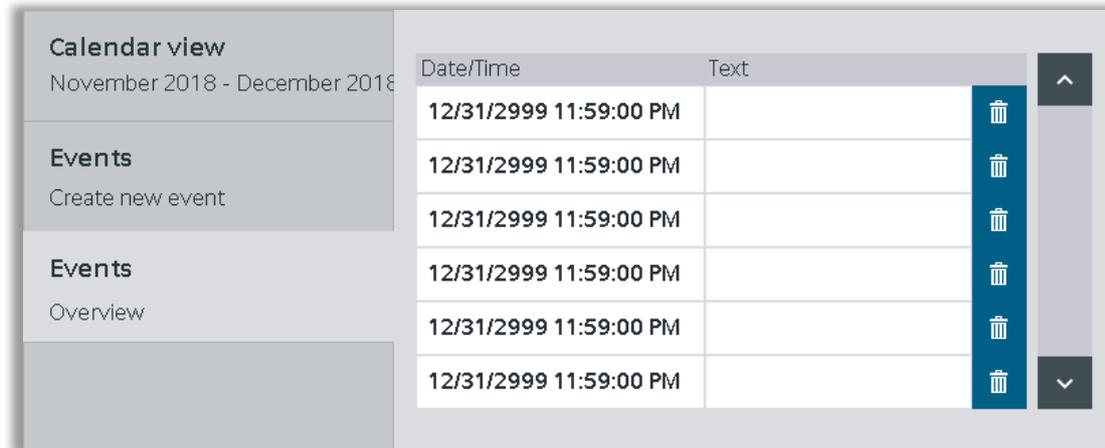
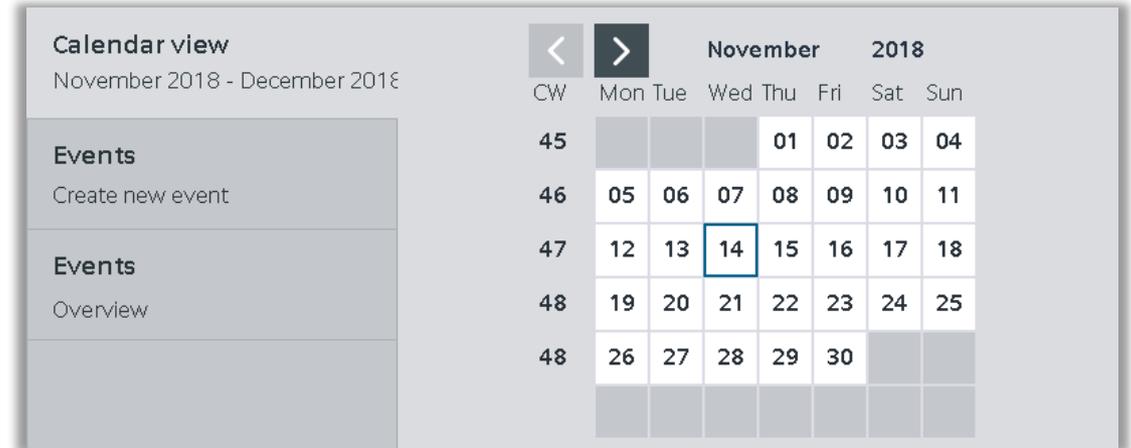
<https://support.industry.siemens.com/cs/ww/de/view/109754400>

# SIMATIC HMI Toolbox

# HMI Toolbox – Zeitfunktionen

## Kalender

- Kalender als Monatsübersicht mit Anzeige der einzelnen Tage des aktuellen und folgenden Monats
- Rote Markierung des aktuellen Tags
- Möglichkeit, verschiedene Termine und Erinnerungen einzutragen



# HMI Toolbox – Daten übertragen

## E-Mails

- E-Mail-Benachrichtigungen zur zeitnahen Reaktionsmöglichkeit
- Automatische E-Mail-Benachrichtigung bei bestimmten Meldungen  
→ Drei-Schicht-System zur richtigen Ansprache des zuständigen Mitarbeiter
- Manuelle Versendung von E-Mails
- Zuverlässige Sicherung bekannter E-Mail-Adressen auch nach Neustart über Rezepturverwaltung

The screenshot shows the 'Auto. E-Mail Versand' settings in the HMI Toolbox. The left sidebar has three items: 'Auto. E-Mail Versand' (highlighted with an orange border), 'Man. E-Mail Versand', and 'Adressbuch'. The main area is titled 'Schichten' and has four tabs: 'Früh', 'Spät', 'Nachts', and 'Einstellungen'. The 'Früh' tab is active, showing a 'Von:' field with '07:00:00' and a 'Bis:' field with '15:00:00'. Below these is an 'E-Mail' field containing 'earlyshift@change.me' and an 'ÖFFNEN' button.

The screenshot shows the 'Man. E-Mail Versand' editor in the HMI Toolbox. The left sidebar has three items: 'Auto. E-Mail Versand', 'Man. E-Mail Versand' (highlighted with an orange border), and 'Adressbuch'. The main area shows an email composition form with the following fields: 'AN ...' with 'earlyshift@change.me', 'Betreff:' with 'Test', and 'Text:' with 'Hallo World'. There is a 'SENDEN' button at the bottom left of the form area.

# HMI Toolbox – Übersicht schaffen

## Datei-Explorer

- Dieser Datei-Explorer ermöglicht erweiterte Dateifunktionen, z.B. Umbenennen, Suchen, Löschen
- Standard-Dateibrowser ermöglicht nur das Verwenden einer beliebigen Datei im Projekt als Variable

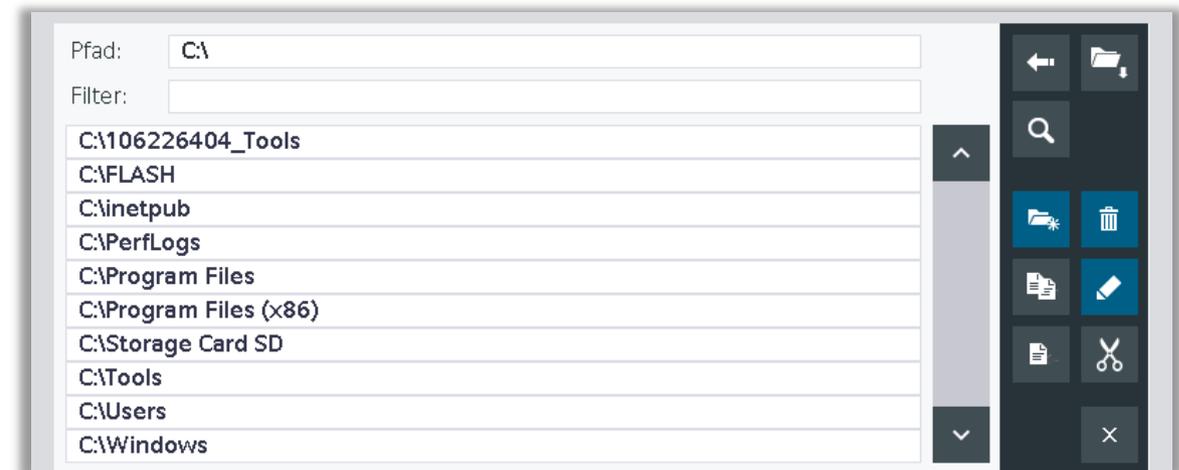
### Neuen Ordner hinzufügen

Name neuer Ordner

**HINZUFÜGEN**

ABBRECHEN



# HMI Toolbox – Download

**SIEMENS**  
*Ingenuity for Life*



Das komplette Anwendungsbeispiel beinhaltet:

- Dokumentation
- TIA Portal Bibliotheken
- TIA Portal Beispiel Projekt

Download unter der Beitrags ID **106226404** oder dem folgenden Link aus dem Industry Online Support:

<https://support.industry.siemens.com/cs/ww/de/view/106226404>

## Informationen



- Objektgruppen als Bildbausteine anlegen
- Stile zur Verwaltung des Aussehens von Objekten
- HMI Template Suite als Vorlage und Inspiration
- HMI Option+ für die volle Kontrolle über das Comfort Panel
- HMI Toolbox – Vorlagen für häufig gebrauchte Funktionen

## Unterlagen



- [HMI Template Suite](#)
- [HMI Design Masterclass](#)
- [HMI Design Workbook](#)
- [Toolbox für HMI Projekte](#)
- [SIMATIC HMI Option+](#)

## Anwendungsfälle



- Erstellung eines neuen HMI
- Erweiterung oder Überarbeitung eines bestehenden HMIs

## SITRAIN - Kurse



- [SIMATIC WinCC maschinennah im TIA Portal \(3 Tage\)](#)
- [SIMATIC WinCC maschinennah im TIA Portal Umsteiger \(E-Learning 6h\)](#)
- [SIMATIC WinCC SCADA im TIA Portal \(5 Tage\)](#)
- [SIMATIC WinCC Unified & Unified Comfort Panels \(3 Tage\)](#)
- [SiVArc - Möglichkeiten der automatischen Visualisierungsgenerierung \(1 Tag\)](#)

# Standardisierungsmatrix

Weiteres Interesse an der Standardisierung?

# Standardisierung in der TIA Portal Umgebung Engineering

Hardwareplanung	Programmierung	Kommunikation	Safety	Engineering
Diagnose	Bedienen und Beobachten	Technologie	Antriebsintegration	
Verteiltes Arbeiten	Versionierung	Test	Maschinen-simulation	Workflow
Benutzerverwaltung	Lizenzmanagement	Software -managemnet	Sicherheit	Administration

# Standardisierungsmatrix in der TIA Portal Umgebung Engineering

	Hardware- planung	Programmierung	Kommunikation	Technologie	Diagnose	Bedienen und Beobachten	Antriebs- integration
Stufe 3	Automatische Projektgenerierung / Automated Engineering Workflow						
	Hardwareprojektierung generieren	Programm generieren	Plug & Communicate	Automatische Generierung	Zentraler Diagnoseserver	Vollständige HMI- Generierung	Antrieb wird automatisch in Betrieb genommen
Stufe 2	Standardisierter Austausch der Konfigurationsdaten	Standardisierte Programmstruktur	Standardisierte Services	Nutzung von Standardanwendungen	Integrierte System- und Prozessdiagnose	Automatische Bildgenerierung	Antrieb wird automatisch generiert
Stufe 1	Toolgestützte Auslegung	Modulare Programmstruktur	Vordefinierte Protokolle	Technologieobjekte	Integrierte Systemdiagnose	Nutzung von Bildbausteinen und Stilen / Designs	Toolgestützte Antriebsprojektierung
Basis	Manuelle Hardwarekonfiguration	Manuelle Programmerstellung	Manuelle Kommunikations- beziehungen	Manuelle Programmierung von Technologiefunktionen	Manuelle Projektierung	Manuelle HMI-Projektierung	Manuelle Antriebsprojektierung

# Standardisierungsmatrix in der TIA Portal Umgebung Engineering Workflow

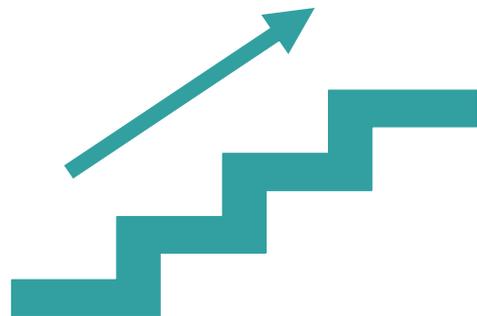
	Verteilte Automation	Verteiltes Arbeiten	Versionierung	Test	Maschinen-simulation
Stufe 3	Plug & Produce für Mehr-Controller-Lösung	Continuous Integration			„Digitalen Zwilling“ automatisch generieren
Stufe 2	Mehr-Controller-Lösung adaptieren bzw. generieren	Unternehmensweiter Projektserver	Unternehmensweites Versionierungssystem	Erweitertes softwarebasiertes Testen	Verknüpfung mit physikalischen Simulationsmodellen
Stufe 1	Mehr-Controller-Lösung manuell	Lokaler Projektserver	Projektweites Versionierungssystem	Einfaches softwarebasiertes Testen	Verwenden von Bibliotheken zur Peripherie-Simulation
Basis	Ein-Controller-Lösung	Manuelles Kopieren & Einfügen in verteilten Projekten	Manuelles Versionieren von Kopiervorlagen & Projekten	Manuelles Testen mit Hardware	Integrierte Simulationsfunktionen nutzen

# Vereinbaren Sie jetzt einen gemeinsamen Termin mit Ihrer Promotion und Fachberatung!

 Wie standardisiert ist mein Unternehmen?

 Wie erreiche ich einen höheren Standardisierungsgrad?

## Standardisierungsmatrix



	Hardware-planung	Programmierung	Kommunikation	Technologie	Diagnose	Bedienen und Beobachten	Antriebs-integration
	Automatische Projektgenerierung / Automated Engineering Workflow						
Stufe 3	Hardwareprojektierung generieren	Programm generieren	Plug & Communicate	Automatische Generierung	Zentraler Diagnoseserver	Vollständige HMI-Generierung	Antrieb wird automatisch in Betrieb genommen
Stufe 2	Standardisierter Austausch der Konfigurationsdaten	Standardisierte Programmstruktur	Standardisierte Services	Nutzung von Standardanwendungen	Integrierte System- und Prozessdiagnose	Automatische Bildgenerierung	Antrieb wird automatisch generiert
Stufe 1	Toolgestützte Auslegung	Modulare Programmstruktur	Vordefinierte Protokolle	Technologieobjekte	Integrierte Systemdiagnose	Nutzung von Bildbausteinen und Stilen / Designs	Toolgestützte Antriebsprojektierung
Basis	Manuelle Hardwarekonfiguration	Manuelle Programmerstellung	Manuelle Kommunikationsbeziehungen	Manuelle Programmierung von Technologiefunktionen	Manuelle Projektierung	Manuelle HMI-Projektierung	Manuelle Antriebsprojektierung

# TIA Portal Optionen

## Arbeiten für und mit Standards

# TIA Portal Optionen

## Arbeiten für und mit Standards

**SIEMENS**  
*Ingenuity for life*



# TIA Portal Optionen @SITRAIN

# SIEMENS

*Ingenuity for Life*

**Openness / SiVArc**

SiVArc (1T)

Openness Prog. 1 (3T)

Openness Prog. 2 (2T)

Automatisches Engineering (5T)

**ProDiag**

V15 Training (Web)

**OPC UA**

Grundlagen S7-1500 (1T)

Grundlagen und Projektierung (4T)

**Test Suite**

V16 Training (Web)

**Standardisierung**

Digital Enterprise Intro – Kompakt (1T)

Einführung in die Standardisierung (1T)

**Multuser Engineering**

V15.1 Training (Web)

V16 Training (Web)

**PLCSIM Advanced**

Virtuelle Inbetriebnahme (5T)

Automatisches Engineering (5T)

**Software Units**

V15.1 Training (Web)

**Version Control Interface (VCI)**

V16 Training (Web)



# TIA Portal Optionen @SIOS

# SIEMENS

*Ingenuity for Life*

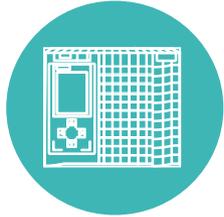
**Openness / SiVArc**  
[SiVArc getting started](#)  
[Openness Handbuch](#)  
[Openness Scripter](#)  
[Openness Demo Basic](#)  
[Openness Demo Adv.](#)

**Standardisierung**  
[Themenseite Bibliotheken](#)  
[Themenseite Standardisierung](#)

**Multuser Engineering**  
[Anwendungsbeispiel](#)



**PLCSIM Advanced**  
[Simulationstabellen](#)  
[Co-Simulation via API](#)  
[S7unitTest](#)



**Software Units**  
[SIMATIC STEP 7](#)  
[Basic/Professional V16](#)  
[Systemhandbuch](#)



**Version Control Interface (VCI)**  
[SIMATIC STEP 7](#)  
[Basic/Professional V16](#)  
[Systemhandbuch](#)



**ProDiag**  
[Anwendungsbeispiel](#)

**OPC UA**  
[SiOME](#)  
[Server – Methoden](#)  
[Client Bausteine](#)  
[Client Lib für MS Excel](#)

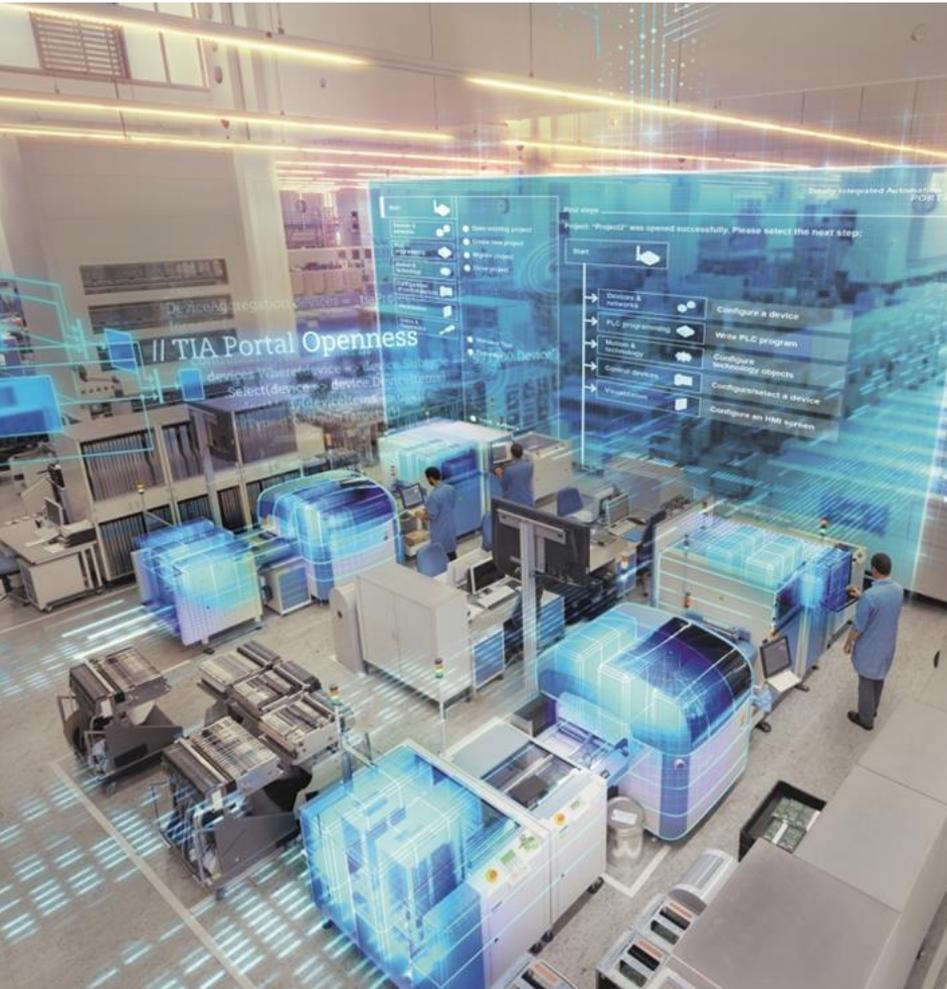


**Test Suite**  
[V16 Training \(Web\)](#)



# Vielen Dank für Ihre Aufmerksamkeit

**SIEMENS**  
*Ingenuity for Life*



**Tobias Vielweber**

Technical Consulting  
RC-DE DI FA 3 FB2

Werner-von-Siemens-Platz 1  
30880 Laatzen  
Deutschland

**Jannes Schlinke**

Sales Specialist  
RC-DE DI FA 2 FB1

Lindenplatz 2  
20099 Hamburg  
Deutschland

Zur Unterstützung / Beratung zum Thema Standardisierung kontaktieren Sie Ihre regionale Fachberatung unter:

[fachberatung.df.pd.de@siemens.com](mailto:fachberatung.df.pd.de@siemens.com)

Tel.: +49 (0)911 /895 -7111